

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

99 175
US (CR)
JCS11 U.S. Pat.
09/651565
08/29/00

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

1999年 9月14日

出 願 番 号

Application Number:

平成11年特許願第260355号

出 願 人

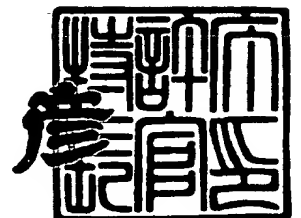
Applicant (s):

インターナショナル・ビジネス・マシーンズ・コーポレイション

1999年11月12日

特許庁長官
Commissioner,
Patent Office

近 藤 隆 彦



出証番号 出証特平11-3079010

【書類名】 特許願

【整理番号】 JA999175

【提出日】 平成11年 9月14日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 12/00

G06F 17/30

【発明者】

【住所又は居所】 神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ビー・エム株式会社 大和事業所内

【氏名】 稲垣 巖

【発明者】

【住所又は居所】 神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ビー・エム株式会社 大和事業所内

【氏名】 濱田 誠司

【発明者】

【住所又は居所】 神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ビー・エム株式会社 大和事業所内

【氏名】 大川 昌弘

【発明者】

【住所又は居所】 神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ビー・エム株式会社 大和事業所内

【氏名】 善家 直己

【発明者】

【住所又は居所】 神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ビー・エム株式会社 大和事業所内

【氏名】 難波 かおり

【特許出願人】

【識別番号】 390009531

【氏名又は名称】 インターナショナル・ビジネス・マシーンズ・コーポレ

イ シ ョ ン

【代理人】

【識別番号】 100086243

【弁理士】

【氏名又は名称】 坂口 博

【復代理人】

【識別番号】 100104880

【弁理士】

【氏名又は名称】 古部 次郎

【選任した代理人】

【識別番号】 100091568

【弁理士】

【氏名又は名称】 市位 嘉宏

【選任した復代理人】

【識別番号】 100100077

【弁理士】

【氏名又は名称】 大場 充

【手数料の表示】

【予納台帳番号】 081504

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9304391

【包括委任状番号】 9304392

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 クライアントサーバーシステム、オブジェクトプール、オブジェクトのプール方法、記憶媒体、およびプログラム伝送装置

【特許請求の範囲】

【請求項 1】 分散オブジェクトを利用したクライアントサーバーシステムであって、

通信ネットワークに接続され、オブジェクトに対するアクセス要求を行うクライアントと、

前記クライアントのアクセス要求に基づいて実オブジェクトによりアプリケーションを実行するアプリケーションサーバーと、

前記クライアントと前記通信ネットワークを介して接続されると共に前記アプリケーションサーバーと接続され、前記実オブジェクトに対応するプロキシオブジェクトをプーリングすると共に、当該実オブジェクトの情報である実オブジェクト管理情報を保持するオブジェクトプールサーバーとを備え、

前記アプリケーションサーバーは前記アプリケーションの状態変化に応じて前記オブジェクトプールサーバーに対してイベントを通知すると共に、当該オブジェクトプールサーバーは当該アプリケーションサーバーからの当該イベントの通知に基づいて前記実オブジェクト管理情報を自動更新することを特徴とするクライアントサーバーシステム。

【請求項 2】 前記アプリケーションサーバーから通知されるイベントは、プロジェクトの開始処理の結果及びプロジェクトの停止処理の結果の少なくとも 1 つの結果に基づいて形成されることを特徴とする請求項 1 記載のクライアントサーバーシステム。

【請求項 3】 分散オブジェクトを用いたオブジェクトプールであって、オブジェクトに対するアクセス要求を解析するクライアント要求解析部と、前記オブジェクトプールの終了処理時にオブジェクト情報を格納するオブジェクト情報記憶部と、

前記オブジェクトプールの開始処理時に前記オブジェクト情報記憶部により格納された前記オブジェクト情報に基づいてオブジェクトを生成するオブジェクト

生成処理部と

前記オブジェクト生成処理部により生成されたオブジェクトを前記クライアントからの当該オブジェクトのアクセス前にプールするオブジェクト管理部とを備えたことを特徴とするオブジェクトプール。

【請求項4】 前記オブジェクト情報記憶部により格納されるオブジェクト情報は、少なくとも最後にアクセスされたオブジェクトであることを認識できるように構成され、

前記オブジェクト生成処理部は、最後にアクセスされた前記オブジェクトから生成を開始することを特徴とする請求項3記載のオブジェクトプール。

【請求項5】 分散オブジェクトを用いたクライアントサーバーシステムであって、

オブジェクトに対するアクセス要求を出すクライアントと通信ネットワークを介して接続され、オブジェクトをプールすると共にオブジェクト情報を管理するオブジェクトプールと、

前記オブジェクトプールに接続され前記クライアントのアクセス要求に基づいてアプリケーションを実行すると共に、当該アプリケーションの状態変化に合わせて前記オブジェクトプールに対してイベントを通知するアプリケーション実行環境とを備え、

前記オブジェクトプールは、前記アプリケーション実行環境からのイベント通知に基づいて前記オブジェクト情報を更新することを特徴とするクライアントサーバーシステム。

【請求項6】 前記オブジェクトプールの機能を有するオブジェクトプールサーバーと、前記アプリケーション実行環境にあるアプリケーションサーバーとがネット等を介して接続されると共に、当該オブジェクトプールサーバーはプロキシオブジェクトとしてのオブジェクトをプールすることを特徴とする請求項5記載のクライアントサーバーシステム。

【請求項7】 前記オブジェクトプールと前記アプリケーション実行環境とは同一のサーバーにて形成されることを特徴とする請求項5記載のクライアントサーバーシステム。

【請求項 8】 分散システムにおけるアプリケーションの実行のために特定のサーバーにて予めオブジェクトをプールするオブジェクトのプール方法であって、

前記サーバーにおけるプロセスの終了処理時にオブジェクト情報を格納するステップと、

前記サーバーにおけるプロセスの開始処理時に前記オブジェクト情報に基づいてオブジェクトを生成するステップと、

生成されたオブジェクトをプールするステップとを含むことを特徴とするオブジェクトのプール方法。

【請求項 9】 前記オブジェクト情報は所定の優先順位を付して格納されると共に、

前記オブジェクトは前記優先順位の高いものから順に生成されることを特徴とする請求項 8 記載のオブジェクトのプール方法。

【請求項 1 0】 オブジェクトに対するアクセス要求を出すクライアントと通信ネットワークを介して接続されると共に、実オブジェクトを実行するアプリケーション実行環境と接続される、オブジェクトプールにおけるオブジェクトのプール方法であって、

前記アプリケーション実行環境における実オブジェクトに対応するプロキシオブジェクトをプールするステップと、

前記アプリケーション実行環境における実オブジェクトの状態変化を認識するステップと、

認識された前記状態変化に基づいて実オブジェクトの管理情報を更新するステップとを備えたことを特徴とするオブジェクトのプール方法。

【請求項 1 1】 コンピュータに実行させるプログラムを当該コンピュータの入力手段が読取可能に記憶した記憶媒体において、

前記プログラムは、分散オブジェクトを利用したアプリケーションを実行するプロジェクトの実行状態を監視する監視処理と、

前記監視処理により監視されたプロジェクトの実行状態に基づいてイベントを生成するイベント生成処理と、

ネット等を介して接続されたオブジェクトプールサーバに対して前記イベント生成処理によって生成されたイベントを発行するイベント発行処理とを前記コンピュータに実行させることを特徴とする記憶媒体。

【請求項 1 2】 分散オブジェクトを利用したアプリケーションを実行するアプリケーションサーバと、当該アプリケーションの実行を要求するクライアントとに対してネット等を介して接続するコンピュータに実行させるプログラムを当該コンピュータの入力手段が読取可能に記憶した記憶媒体において、

前記プログラムは、

前記アプリケーションサーバが実行する実オブジェクトに対応するプロキシオブジェクトをプールするオブジェクトプール処理と、

前記アプリケーションサーバから発行されるイベントを受信すると共に、当該イベントに基づいて前記実オブジェクトの管理情報を更新する更新処理とを前記コンピュータに実行させることを特徴とする記憶媒体。

【請求項 1 3】 前記オブジェクトの実行状態を把握してオブジェクト毎に管理する実行状態管理処理と、

前記クライアントからのオブジェクト生成要求および/またはオブジェクト削除要求を含むリクエストを解析するリクエスト解析処理と、

前記リクエスト解析処理によるリクエストの解析結果を受け、前記実行状態管理処理による管理結果に基づいてオブジェクトの生成および/またはオブジェクトの削除を実行する処理とを更に前記コンピュータに実行させることを特徴とする請求項 1 2 記載の記憶媒体。

【請求項 1 4】 コンピュータに、分散オブジェクトを利用したアプリケーションの状態変化に応じてイベントを形成するイベント形成処理プログラムと、当該イベント形成処理により形成されたイベントに基づいてオブジェクトをプールするオブジェクトプール処理プログラムとを実行させるソフトウェアプロダクトを記憶する記憶手段と、

前記記憶手段から前記プログラムを読み出して前記ソフトウェアプロダクトを送信する送信手段とを備えたことを特徴とするプログラム伝送装置。

【請求項 1 5】 コンピュータに、分散オブジェクトを利用したアプリケー

ションの実行に伴うオブジェクトをサーバーにてプールするオブジェクトプール処理と、当該サーバーにおいてオブジェクト情報を格納する情報格納処理と、当該情報格納処理により格納された当該オブジェクト情報に基づいて生成するオブジェクトの順序を決定する生成順序決定処理を実行するプログラムを記憶する記憶手段と、

前記記憶手段から前記プログラムを読み出して当該プログラムを送信する送信手段とを備えたことを特徴とするプログラム伝送装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、分散オブジェクトを使用したアプリケーションを実行するクライアントサーバーシステムに関する。

【0002】

【従来の技術】

近年、ウェブ(Web)ブラウザをベースとしたアプリケーション形態が急速に増加しており、分散オブジェクトの利用方法が大きく変わってきている。具体的には、以前はクライアントから直接、サーバー上のオブジェクトを生成する方法が主体であったが、現在は、Webサーバー若しくはWebアプリケーションサーバー上で分散オブジェクト技術を利用して、他のサーバー上のオブジェクトを生成するアプリケーションモデルが普及してきている。実際の実装例としては、マイクロソフト社のASPとMTSとの組合せや、出願人(IBM社)のWebSphereとComponentBrokerの組み合わせが挙げられる。

【0003】

ここで、分散オブジェクトとは、データ、アプリケーション、プロセスなどを部品化したものであり、これらのオブジェクトは場所に関係なく、ネットワークのどこからでもアクセスでき、業務に使用することが可能である。より詳細には、オブジェクト指向におけるメソッド呼び出しを遠隔化できるようにしたものであり、ソケットレベルの上位層にオブジェクト指向的なプログラミングインターフェイスを実現するミドルウェアとして構築されたものである。この分散オブジ

ェクトのプログラミング技術では、離れたマシンにあるオブジェクトがまるで蜃気楼のように自マシンに現れ、それを操作することで遠くにあるオブジェクトを自由に操作することができる。この分散オブジェクトでは、蜃気楼のように現れるオブジェクトをプロキシオブジェクト(代理オブジェクト)と呼び、離れたマシンにあるオブジェクト(実オブジェクト: リモートオブジェクト)と見かけ上は同じであるが、本物ではなく代理として存在させており、クライアントから最初にアクセスするポイントは、実際のサーバーではなくWebサーバーとなる形態である。

【0004】

この分散オブジェクト環境を使用した場合に、ソフトウェア的に最もコストがかかるのはオブジェクトの作成処理である。実際に分散オブジェクトのリモートオブジェクトの生成処理は、他の処理に比べて時間がかかることが知られており、オブジェクト作成時の処理をいかに早くするかが全体のレスポンスタイムを短くする鍵となる。このとき、特開平10-11339号公報に記載されているようなデータベース管理システムにおけるオブジェクトプーリング技術を転用し、クライアントからのアクセス前に予めオブジェクトを作成しておき、クライアントからオブジェクトの作成要求があったときに、そのプールされたところからオブジェクトを渡すことができれば、結果としてレスポンスタイムの短縮を図ることが可能となる。

【0005】

【発明が解決しようとする課題】

しかしながら、上記公報に記載されているような既存技術のプーリングは、クライアントからのリクエストを受け、利用し終わったものをプールするメカニズムであるため、サーバー起動時にはプーリングメカニズムが有効とはならず、決められたレスポンスタイムを実現することは困難である。また、既存技術のようなクライアントから来るであろう要求が明確であるデータベースシステムでは、初期化の際に単純に予め決められたオブジェクトを生成することが可能であるが、データベースのような決まりきったオブジェクトではなく、分散オブジェクトのような実行時でないと作られるオブジェクトが解らないような環境下では、既

存のプーリングメカニズムを実装するだけでは十分な効果を得ることができない。

【0006】

また、通常のオブジェクトプールでは、生成されたオブジェクトに対する開放要求でプーリングし、次に同一クラスに対する生成要求によりプール中にあるオブジェクトをクライアントに割り当てるように構成することでパフォーマンスの向上を企図しているが、必ずしも有効なオブジェクトが常にプールされているとは限らない。特に分散オブジェクトを利用したアプリケーションにてプロキシオブジェクトのプーリングを行う場合には、クライアントからのオブジェクト生成要求に対して、実際のサーバー上では既に無効になってしまったプロキシオブジェクトを割り当ててしまう可能性がある。かかる不整合が生じた場合には、例えばクライアントが無効になってしまったオブジェクトを使用するとエラーが発生してしまい、オブジェクトプールの意味が無くなる。オブジェクト毎に有効期限を設定し、一定期間使用されなかったオブジェクトを自動的に削除することでオブジェクトの有効性を維持する方法も考えられるが、この機能だけでは実際に有効なオブジェクトもプールから削除される危険性が残る。

【0007】

この問題点を考慮して、オブジェクトを開放する際にある特定のメソッドを呼び出し、このメソッドの中でユーザーが実装したロジックに基づいて、そのオブジェクトをプールできるか否かを制御できる機能を提供するオブジェクトプールの実装形態もある。この方法は、ユーザーが自由にオブジェクトの動きを制御できる点で柔軟性に富んでいる。しかしながら、一方で、ユーザー自身に対してプログラム作成を強いることとなりユーザーの負担が大きくなると共に、常にそのロジックを通ることとなり処理量が増してしまう。また、このようにオブジェクトの実装としてプール使用時の振る舞いを埋め込む方法では、アプリケーション起動時などのオブジェクトが存在しない期間の動作を制御することが困難となる。

【0008】

本発明は以上のような技術的課題を解決するためになされたものであって、サ

ーバーの起動後、クライアントから最初にオブジェクトの生成要求があったときの応答時間を短縮することを目的とする。

更に、アプリケーションとオブジェクトプールとの同期を取ることで、オブジェクトプールの有用性を向上させることを他の目的とする。

【0009】

【課題を解決するための手段】

かかる目的のもと、本発明は、分散オブジェクトを利用したクライアントサーバーシステムであって、通信ネットワークに接続され、オブジェクトに対するアクセス要求を行うクライアントと、このクライアントのアクセス要求に基づいて実オブジェクトによりアプリケーションを実行するアプリケーションサーバーと、クライアントと通信ネットワークを介して接続されると共にこのアプリケーションサーバーと接続され、実オブジェクトに対応するプロキシオブジェクトをプーリングすると共に、実オブジェクトの情報である実オブジェクト管理情報を保持するオブジェクトプールサーバーとを備え、このアプリケーションサーバーはアプリケーションの状態変化に応じてオブジェクトプールサーバーに対してイベントを通知すると共に、オブジェクトプールサーバーはアプリケーションサーバーからのイベントの通知に基づいて実オブジェクト管理情報を自動更新することを特徴としている。

【0010】

このアプリケーションサーバーから通知されるイベントは、プロジェクトの開始処理の結果及びプロジェクトの停止処理の結果の少なくとも1つの結果に基づいて形成されることを特徴とすれば、実際には使用不可能なオブジェクトをクライアントに割り当ててしまう危険性を排除することができる点からも好ましい。

このプロジェクトの停止処理は、アプリケーションサーバーのユーザーによるプロジェクトの停止要求に係わるものの他、プロジェクトの異常終了による停止処理をも含むものである。

また、このアプリケーションサーバーとオブジェクトプールサーバーとは必ずしも異なった筐体である必要はなく、同一の筐体に属するものであっても構わない。

【0011】

また、本発明は、分散オブジェクトを用いたオブジェクトプールであって、オブジェクトに対するアクセス要求を解析するクライアント要求解析部と、オブジェクトプールの終了処理時にオブジェクト情報を格納するオブジェクト情報記憶部と、オブジェクトプールの開始処理時にオブジェクト情報記憶部により格納されたオブジェクト情報に基づいてオブジェクトを生成するオブジェクト生成処理部と、このオブジェクト生成処理部により生成されたオブジェクトをクライアントからのオブジェクトのアクセス前にプールするオブジェクト管理部とを備えたことを特徴としている。

【0012】

このオブジェクト情報記憶部により格納されるオブジェクト情報は、少なくとも最後にアクセスされたオブジェクトであることを認識できるように構成され、このオブジェクト生成処理部は、最後にアクセスされたオブジェクトから生成を開始することを特徴とすれば、最終アクセスがなされたオブジェクトは再度、利用される可能性が高いことから、オブジェクトプールとして特に効果的である。

ここで、このオブジェクト情報として、優先順位が付与されるように構成すれば、動的にオブジェクトをプールすることができる点で好ましい。この優先順位としては、最終アクセス時間の他、アクセス回数(頻度)等、実装の形態によって任意に選択することが可能である。

【0013】

また、本発明の分散オブジェクトを用いたクライアントサーバーシステムは、オブジェクトに対するアクセス要求を出すクライアントと通信ネットワークを介して接続され、オブジェクトをプールすると共にオブジェクト情報を管理するオブジェクトプールと、このオブジェクトプールに接続されクライアントのアクセス要求に基づいてアプリケーションを実行すると共に、アプリケーションの状態変化に合わせてオブジェクトプールに対してイベントを通知するアプリケーション実行環境とを備え、オブジェクトプールは、このアプリケーション実行環境からのイベント通知に基づいてオブジェクト情報を更新することを特徴としている。

【0014】

ここで、オブジェクトプールの機能を有するオブジェクトプールサーバーと、アプリケーション実行環境にあるアプリケーションサーバーとがネット等を介して接続されると共に、このオブジェクトプールサーバーはプロキシオブジェクトとしてのオブジェクトをプールすることを特徴とすることもできる。

また、オブジェクトプールとアプリケーション実行環境とは同一のサーバーにて形成されることを特徴とすることも可能である。

【0015】

また、本発明は、分散システムにおけるアプリケーションの実行のために特定のサーバーにて予めオブジェクトをプールするオブジェクトのプール方法であって、サーバーにおけるプロセスの終了処理時にオブジェクト情報を格納するステップと、このサーバーにおけるプロセスの開始処理時にオブジェクト情報に基づいてオブジェクトを生成するステップと、生成されたオブジェクトをプールするステップとを含むことを特徴としている。

【0016】

ここで、オブジェクト情報は所定の優先順位を付して格納されると共に、オブジェクトは優先順位の高いものから順に生成されることを特徴とすれば、お客様のレスポンスタイムに対する一定した要求を満たすことが可能となる点で好ましい。

この優先順位の用い方としては、例えば管理している空きオブジェクトの数が最大プール数に達していたら優先順位の最も低いオブジェクトをプールから外し、実際に削除する等に用いることが可能である。また、サーバープロセスの開始処理時に終了時に保存したデータから優先順位の高い順に、初期プール数に達するまでオブジェクトを生成するように構成すれば、クライアントが使用するであろうオブジェクトの決定を動的に行うことが可能となり、クライアントの要求に対する応答時間の短縮を図ることができる点で優れている。

【0017】

また、本発明は、オブジェクトに対するアクセス要求を出すクライアントと通信ネットワークを介して接続されると共に、実オブジェクトを実行するアプリケ

ーション実行環境と接続される、クライアントサーバーシステムにおけるオブジェクトのプール方法であって、このアプリケーション実行環境における実オブジェクトに対応するプロキシオブジェクトをプールするステップと、実オブジェクトの管理情報を記憶するステップと、アプリケーション実行環境における実オブジェクトの状態変化を認識するステップと、この認識された状態変化に基づいて記憶されている実オブジェクトの管理情報を更新するステップとを備えたことを特徴としている。特にこの状態変化を非同期のシグナルであるイベントにより認識できるように構成すれば、最小限のコストでサーバー上のアプリケーションとオブジェクトプール内のオブジェクトとの同期を取ることが可能となる点で好ましい。

【0018】

また、本発明は、コンピュータに実行させるプログラムをこのコンピュータの入力手段が読取可能に記憶した記憶媒体において、このプログラムは、分散オブジェクトを利用したアプリケーションを実行するプロジェクトの実行状態を監視する監視処理と、この監視処理により監視されたプロジェクトの実行状態に基づいてイベントを生成するイベント生成処理と、このイベント生成処理によって生成されたイベントを発行するイベント発行処理とをこのコンピュータに実行させることを特徴としている。

ここで、このイベント発行処理は、ネット等を介して接続されたオブジェクトプールサーバに対してイベントを発行することを特徴とすれば、例えばアプリケーションを実行するアプリケーションサーバーにてこの記憶媒体を用いて一連の処理を入力することが可能となり、最小限のコストでサーバー上のアプリケーションとオブジェクトプール内のオブジェクトとの同期を取ることが可能とする点で優れている。尚、記憶媒体としてはCD-ROMやフロッピーディスク等、その種類が問われることはなく、そのソフトウェアが販売される媒体としての記憶媒体を含むものである。

【0019】

また、本発明は、分散オブジェクトを利用したアプリケーションを実行するアプリケーションサーバーと、アプリケーションの実行を要求するクライアントと

に対してネット等を介して接続するコンピュータに実行させるプログラムをコンピュータの入力手段が読取可能に記憶した記憶媒体において、このプログラムは、アプリケーションサーバーが実行する実オブジェクトに対応するプロキシオブジェクトをプールするオブジェクトプール処理と、アプリケーションサーバーから発行されるイベントを受信すると共に、このイベントに基づいて実オブジェクトの管理情報を更新する更新処理とをコンピュータに実行させることを特徴としており、オブジェクトプールサーバーに対してこのソフトウェアを容易に実行させることが可能となる点で優れている。

更に、このオブジェクトの実行状態を把握してオブジェクト毎に管理する実行状態管理処理と、このクライアントからのオブジェクト生成要求および/またはオブジェクト削除要求を含むリクエストを解析するリクエスト解析処理と、このリクエスト解析処理によるリクエストの解析結果を受け、実行状態管理処理による管理結果に基づいてオブジェクトの生成および/またはオブジェクトの削除を実行する処理とを更にコンピュータに実行させることを特徴とすれば、アプリケーションサーバー上で実行されるアプリケーションの状態変化を適切に把え、プール内のオブジェクトを自動的に更新することが可能となる点で好ましい。尚、記憶媒体の種類を問わない点は前述と同様である。

【0020】

また、本発明をプログラム伝送装置に適用すると、コンピュータに、分散オブジェクトを利用したアプリケーションの状態変化に応じてイベントを形成するイベント形成処理プログラムと、このイベント形成処理により形成されたイベントに基づいてオブジェクトをプールするオブジェクトプール処理プログラムとを実行させるソフトウェアプロダクトを記憶する記憶手段と、この記憶手段からプログラムを読み出してこのソフトウェアプロダクトを送信する送信手段とを備えたことを特徴としている。

また、本発明のプログラム伝送装置は、コンピュータに、分散オブジェクトを利用したアプリケーションの実行に伴うオブジェクトをサーバーにてプールするオブジェクトプール処理と、サーバーにおいてオブジェクト情報を格納する情報格納処理と、この情報格納処理により格納されたオブジェクト情報に基づいて生

成するオブジェクトの順序を決定する生成順序決定処理を実行するプログラムを記憶する記憶手段と、この記憶手段からプログラムを読み出してプログラムを送信する送信手段とを備えたことを特徴としている。このようなプログラム伝送装置により、プログラムの提供形態としてCD-ROM等の記憶媒体を介することなく、お客様に対して本発明の技術を提供することが可能となる。

【0021】

【発明の実施の形態】

[実施の形態1]

以下、添付図面に示す実施の形態に基づいてこの発明を詳細に説明する。

図1は、本実施の形態におけるクライアントサーバーシステムの全体構成を説明するための図である。同図において、符号11はクライアントであり、オブジェクトの生成要求や削除要求等の各種リクエストを出しており、クライアントプログラムはオブジェクトへのメソッドの呼び出しを行うという形でサービスの実行を要求している。また、符号20はアプリケーション実行環境としてのアプリケーションサーバーである。また、符号30はオブジェクトプールであり、広域ネットワークであるWAN(Wide Area Network)や構内ネットワークであるLAN(Local Area Network)によってクライアント11と接続され、更にアプリケーションサーバー20と接続されている。ここで、アプリケーションサーバー20には実オブジェクト16が、アプリケーションサーバー20にはこの実オブジェクト16に対する代理オブジェクトとしてのプロキシオブジェクト17が存在するように構成されている。

【0022】

図2は、アプリケーション実行環境であるアプリケーションサーバー20の機能ブロックを説明するための図である。同図に示すプロジェクトは、一まとまりのアプリケーションの実行単位であり、単数または複数のオブジェクトによってアプリケーションが実行できるように構成されている。符号21はプロジェクトの実行状況に応じてイベントを作成すると共にオブジェクトプール30のイベント処理部31(後述)に対してイベントを発行するイベント生成部、22はアプリケーションサーバー20のオペレータであるユーザーからのリクエスト内容を解

析するユーザー入力処理部である。ここで、このイベントとは、イベントのサブライヤがある特定の事象をイベントのコンシューマに伝えるための非同期のシグナルのことである。また、23はユーザー入力処理部22の要求によってプロジェクトの起動を開始するプロジェクト開始処理部、24はプロジェクトの終了処理を実行するプロジェクト停止処理部である。更に、25はプロジェクトの実行を監視すると共にイベント生成部21に対してイベント作成を依頼するプロジェクト実行監視部である。また、26はプロジェクトの前処理や後処理を実行し、各プロジェクト間を全体として制御するプロジェクト実行制御部である。

【0023】

図3は、イベント生成部21において生成されるイベント通知管理テーブルを示すものである。このイベント通知管理テーブルは、オブジェクトプール30のIDであるオブジェクトプールIDと、その名称であるホスト名、およびイベントフィルタの対応付けが可能となるテーブルを構成している。このイベントフィルタは、アプリケーションにおいてどのような状態変化が生じたときにどのオブジェクトプール30に対してイベントを発行するか、を示すものであり、例えば「All」であれば全ての状態変化が生じたときにイベントを発行し、「Only Error」であればエラー状態のときに、また、「Warning」であれば予期しない警告状態のときにイベントを発行するように管理される。このイベント管理テーブルに登録された情報に基づいて、イベント生成部21は、所定の状態変化時に所定のオブジェクトプール30に対してイベントを発行している。

【0024】

図4は、プロジェクト実行監視部25において更新されるプロジェクト実行監視テーブルを示すものである。このプロジェクト実行監視テーブルは、アプリケーションサーバー20にて実行されるプロジェクトのIDであるプロジェクトIDとその名称であるプロジェクト名、およびそのプロジェクトにおける実行状態を対応付けしたテーブルである。このプロジェクト実行監視テーブルは、プロジェクト開始処理部23からのプロジェクトの開始処理の結果に基づいて更新されると共に、この更新されたプロジェクト実行監視テーブルに基づいてイベント生

成部 21 によるイベントの発行が可能となるように構成されている。

【0025】

図5は、オブジェクトプール30の機能ブロックを説明するための図である。同図において、符号31はイベント処理部、32はオブジェクト管理部であり、イベント処理部31は、アプリケーションサーバー20のイベント生成部21から通知されたイベントに基づき、オブジェクト管理部32に対してオブジェクト実行状態の変更を通知している。また、33は活動中オブジェクト管理部であり、既にクライアント11に割り当てられているオブジェクトを管理している。更に、34は空きオブジェクト管理部であり、未だクライアント11に割り当てられていないオブジェクトを管理している。また、35はハードディスク等からなるオブジェクト情報記憶部であり、例えばオブジェクトプール30の終了時にオブジェクト情報を読み取って記憶している。また、36は開始処理部、37は終了処理部であり、それぞれ、オブジェクトプール30の開始時(イニシャル時)や終了時にオブジェクト情報記憶部35からのオブジェクト情報の読み取りや格納を実施している。

【0026】

また、図5において、符号41はクライアント11からのリクエストの内容を解析するクライアント要求解析部、42はクライアント要求解析部41からの依頼に基づいてオブジェクト管理部32に対してオブジェクトの取得要求を行うオブジェクト生成処理部である。また、43はメソッドコール処理部であり、クライアント11からメソッドコール処理要求があった場合にクライアント要求解析部41からの依頼に基づいてオブジェクト管理部32に対して処理要求のあったオブジェクトを請求している。更に、44はクライアント要求解析部41からの依頼に基づいてオブジェクト管理部32に対してオブジェクトの削除要求を行うオブジェクト削除処理部である。クライアント要求解析部41が受けるクライアント11からのリクエストは、例えば、特定のクライアント11がWebのホームページを開いた時点、即ちホームページをダウンロードした時点で、所定のプログラムが呼び込まれ、オブジェクトの生成要求が発行されてWebページ上に例えばアプレット(Applet)が動作した後、サーバー上のオブジェクトプール30

に対してオブジェクトの生成要求をするように構成することができる。このクライアント 11 からのリクエストは、実装の仕方によって変わるもので、例えば後述する図 22 に示すオーダーフォーム (Order Form) の画面上にオブジェクト生成ボタンを設けるような形態にて実装することも可能である。

【0027】

図 6 は、活動中オブジェクト管理部 33 において用いられる活動中オブジェクト管理テーブルを示すものである。このテーブルにおいて、オブジェクト ID はクライアント 11 に対して既に割り当てられているオブジェクトの ID を示している。また、クラス名はオブジェクトの機能や性質を定義する所謂クラスをオブジェクト毎に示したものである。また、サーバー名はオブジェクトが実行されるアプリケーションサーバー 20 をオブジェクト単位で管理するものであり、プロジェクト名は一まとまりのアプリケーションの実行単位であるプロジェクトを対応付けして管理している。また、実行状態の項目は、クライアントに割り当てられた後のオブジェクトの実行状態を管理するものであり、イベント処理部 31 より通知されたプロジェクトに属するオブジェクトを、オブジェクト管理部 32 が選択し、更新した内容が登録される。

【0028】

図 7 は、空きオブジェクト管理部 34 にて用いられる空きオブジェクト管理テーブルを示すものであり、オブジェクトは作成されているがどのクライアントも使用していないオブジェクトがこのテーブルにて管理される。このテーブルにおける優先順位は、オブジェクトの最終アクセス時間などによって決定されるもので、例えば終了処理部 37 によるオブジェクトプールの終了処理時に優先順位を付加した状態でオブジェクト情報記憶部 35 にオブジェクト情報を保存する際や、オブジェクトプールの開始処理時にその高い順に初期プールする等、分散オブジェクト環境下におけるダイナミックなオブジェクトプール形成に際して重要な項目である。また、この優先順位は、空きオブジェクトの数が最大プール数に達した場合に、その一番低いオブジェクトから削除する等にも用いられる。また、オブジェクト ID は作成されてはいるがクライアント 11 に割り当てられていないオブジェクトの ID を示し、クラス名はそのオブジェクトのクラスを示したも

のである。また、サーバー名はオブジェクトが実行されるアプリケーションサーバー 20 を管理し、プロジェクト名はそのプロジェクトを対応付けして管理している。また、実行状態は、アプリケーションサーバー 20 上での実行状態を示している。この実行状態の管理としては、単に停止中だけではなく、回復処理中や一時停止中等の細かな状態管理を実行することも可能である。

【0029】

図 8 は、オブジェクトプールの各種情報を管理するためのオブジェクトプールプロパティ管理テーブルを示している。このテーブルでは、プロパティとして最大プール数と初期プール数が管理されており、この値により、プールされているオブジェクトの削除やイニシャル時におけるオブジェクトの作成を動的に制御することが可能となる。

また、図 9 は、オブジェクト情報記憶部 35 に記憶されるプールオブジェクト情報を示すものであり、優先順位と共にオブジェクトのクラス名、実行するアプリケーションサーバー 20 のサーバー名、そのサーバーにて行われるプロジェクト名がそれぞれ格納される。この格納された情報に基づいて、開始処理時であるイニシャル時に動的なオブジェクトプール作業が実行される。

本実施の形態では、このように各種テーブル構造によりオブジェクト情報を管理することで、データ・モデルが理解し易く、検索速度を大幅に高速化することが可能となる。

【0030】

次に、アプリケーション実行環境であるアプリケーションサーバー 20 におけるプロジェクトの開始動作について、図 10 のフローチャートを用いて説明する。

まず、ユーザー入力処理部 22 は、アプリケーションサーバー 20 のユーザーにより入力されたリクエストの内容を解析し、プロジェクト開始処理部 23 に対してプロジェクトの開始を要求する(ブロック 101)。この開始要求を受け、プロジェクト開始処理部 23 は、プロジェクトの起動を開始すると共に、プロジェクト実行監視部 25 に対してプロジェクトが開始処理に入ったことを通知する(ブロック 102)。この開始処理に入ると、プロジェクト実行監視部 25 は、プ

プロジェクトが準備中であることを示すプロジェクト情報を前述のプロジェクト実行監視テーブルに登録すると同時に、イベント生成部 21 に対してイベントの生成を依頼する(ブロック 103)。イベント生成部 21 は、前述のイベント通知管理テーブルに登録された情報に基づき、イベント発行を要するオブジェクトプール 30 のイベント処理部 31 に対して、プロジェクト準備中のイベントを発行する(ブロック 104)。一方、プロジェクト開始処理部 23 は、プロジェクトの開始処理が完了した時点で、その結果をプロジェクト実行監視部 25 に通知する(ブロック 105)。プロジェクト実行監視部 25 は、プロジェクト開始処理部 23 から受け取ったこのプロジェクトの開始処理の結果をもとに、プロジェクト実行監視テーブルを更新すると共に、イベント生成部 21 に対してイベントの作成を依頼する(ブロック 106)。イベント生成部 21 は、イベント通知管理テーブルに登録された情報に基づき、オブジェクトプール 30 のイベント処理部 31 に対して、プロジェクト実行中(プロジェクトの開始処理が成功した場合)あるいはプロジェクト停止中(プロジェクトの開始処理が不成功の場合)のイベントを発行し(ブロック 107)、プロジェクトの開始動作が終了する。

【0031】

次に、アプリケーションサーバー 20 におけるプロジェクトの停止動作について、図 11 のフローチャートを用いて説明する。

まず、ユーザー入力処理部 22 は、アプリケーションサーバー 20 のユーザーが入力したリクエストの内容を解析し、プロジェクト停止処理部 24 に対して、プロジェクトの停止を要求する(ブロック 111)。プロジェクト停止処理部 24 は、この要求を受けてプロジェクトの停止処理を開始すると共に、プロジェクト実行監視部 25 に対して、プロジェクトが停止処理に入ったことを通知する(ブロック 112)。この通知を受けたプロジェクト実行監視部 25 は、前述したプロジェクト実行監視テーブルに登録されているプロジェクトの実行状態を停止処理中に更新すると共に、イベント生成部 21 に対してイベントの生成を依頼する(ブロック 113)。この依頼を受けたイベント生成部 21 は、イベント通知管理テーブルに登録された情報に基づき、オブジェクトプール 30 のイベント処理部 31 に対して、プロジェクト停止処理中のイベントを発行する(ブロック 114)

。プロジェクト停止処理部 24 は、プロジェクトの停止処理が完了した時点で、その結果をプロジェクト実行監視部 25 に通知する(ブロック 115)。この通知を受けたプロジェクト実行監視部 25 は、プロジェクト実行監視テーブルに登録されているプロジェクトの実行状態を停止中に更新すると共に、イベント生成部 21 に対してイベントの作成を依頼する(ブロック 116)。この依頼を受けたイベント生成部 21 は、イベント通知管理テーブルに登録された情報に基づき、オブジェクトプール 30 のイベント処理部 31 に対してプロジェクト停止中のイベントを発行し(ブロック 117)、一連の停止動作が終了する。

【0032】

次に、アプリケーションサーバー 20 におけるプロジェクトの異常終了処理動作について、図 12 のフローチャートを用いて説明する。

何らかの理由によりプロジェクトの異常終了があったとき、プロジェクト実行監視部 25 は、プロジェクトの異常終了を検知し、プロジェクト停止処理部 24 に対してプロジェクトの異常終了処理の開始を要求する。同時に、前述したプロジェクト実行監視テーブルに登録されているプロジェクトの実行状態を異常終了処理中に更新すると共に、イベント生成部 21 に対してイベントの生成を依頼する(ブロック 121)。イベント生成部 21 は、イベント通知管理テーブルに登録された情報に基づき、オブジェクトプール 30 のイベント処理部 31 に対して、プロジェクト異常終了処理中のイベントを発行する(ブロック 122)。一方、プロジェクト停止処理部 24 は、プロジェクトの異常終了処理が完了した時点で、その結果をプロジェクト実行監視部 25 に通知する(ブロック 123)。この通知を受けたプロジェクト実行監視部 25 は、プロジェクト実行監視テーブルに登録されているプロジェクトの実行状態を停止中に更新すると共に、イベント生成部 21 に対してイベントの作成を依頼する(ブロック 124)。この依頼を受けたイベント生成部 21 は、イベント通知管理テーブルに登録された情報に基づき、オブジェクトプール 30 のイベント処理部 31 に対して、プロジェクト停止中のイベントを発行し(ブロック 125)、プロジェクトの異常終了処理動作が終了する。

【0033】

以上のように、アプリケーション実行環境であるアプリケーションサーバー 20 にてイベントが形成されるが、このイベントを受けるオブジェクトプール 30 でのイベント受信時の動作について図 13 の動作フロー図を用いて説明する。尚、オブジェクトプール 30 においてプールされるオブジェクトはプロキシオブジェクト 17 であるが、便宜上、オブジェクトの文言にて説明する。

オブジェクトプール 30 におけるイベント処理部 31 は、アプリケーションサーバー 20 のイベント生成部 21 から通知されたイベントに基づき、オブジェクト管理部 32 に対してオブジェクト実行状態の変更を通知する(ブロック 131)。この通知を受けたオブジェクト管理部 32 は、活動中オブジェクト管理部 33、空きオブジェクト管理部 34 に対してそれぞれのオブジェクト管理テーブルに登録されたオブジェクトのうち、イベント処理部 31 より通知されたプロジェクトに属するオブジェクトを選択し、その実行状態を更新する(ブロック 132)。

【0034】

このように本実施の形態によれば、アプリケーションサーバー 20 にて実行されているアプリケーションの状態が変化した際に、アプリケーションサーバー 20 自身がオブジェクトプール 30 に対してイベント通知を行うことにより、オブジェクトプール 30 にてアプリケーションサーバー 20 の状態を把握することができ、プールされたオブジェクトを自動的に更新することが可能となる。その結果、アプリケーションサーバー 20 で実行されているアプリケーションの状態変化に迅速に対応することができ、実際には使用不可能なオブジェクトをクライアント 11 に割り当ててしまう危険性を回避することが可能となる。言い換えれば、本実施の形態によるオブジェクトプール 30 は、ただ単にオブジェクトをプーリングするのではなく、アプリケーションサーバー 20 上のアプリケーションの状態を把握することで、常に有効なオブジェクトをプールしておくことが可能となる。そのため、仮にクライアント 11 から、ある時点で無効になってしまったオブジェクトに対するアクセス要求があった場合でも、オブジェクトプール 30 からクライアント 11 に対して現在のオブジェクトの情報を直接通知することができ、アプリケーションサーバー 20 へのアクセスなどの冗長な処理を省略でき、オブジェクトプール 30 をより効果的に活用することができる。

【0035】

次に、本実施の形態におけるオブジェクトプール30の開始動作(イニシャル時)について、図14のフローチャートを用いて説明する。

まず、オブジェクトプール30の開始処理部36は、終了時に保存したプールオブジェクト情報をオブジェクト情報記憶部35から読み取り、オブジェクトに対して割り当てられた優先順位に基づき、オブジェクト管理部32に対してオブジェクトの生成要求を行う(ブロック141)。この生成要求を受けたオブジェクト管理部32は、開始処理部36からの生成要求に従い、該当するオブジェクトを作成し、生成されたオブジェクトの情報を空きオブジェクト管理部34に通知する(ブロック142)。この通知を受けた空きオブジェクト管理部34は、オブジェクト管理部32より受け取った情報に基づき、前述した空きオブジェクト管理テーブルに作成されたオブジェクトを登録する(ブロック143)。

【0036】

一方、オブジェクトプール30の終了動作について、図15のフローチャートを用いて説明する。

まず、オブジェクトプール30の終了処理部37は、オブジェクト管理部32及びクライアント要求解析部41に対して終了処理の開始を通知する(ブロック151)。クライアント要求解析部41は、この終了処理開始の通知を受け取ると、以降クライアント11からの処理要求を拒否する(ブロック152)。次に、オブジェクト管理部32は、空きオブジェクト管理部34から、空きオブジェクト管理テーブルにあるオブジェクト情報を取得し、終了処理部に転送する(ブロック153)。終了処理部37は、オブジェクト管理部32から受け取ったオブジェクト情報を、プロセス終了後も消去されないよう、オブジェクト情報記憶部35に格納する(ブロック154)。この格納されるオブジェクト情報は、例えば前述の図9に示すような内容であり、この図9に示す優先順位に基づいて図14にて説明した開始動作が実行され、例えば優先順位の高いものから順にオブジェクトが生成される。

【0037】

以上説明したように、本実施の形態におけるオブジェクトプール30の終了動

作および開始動作によれば、データベースのような決まりきったものではない分散オブジェクトにおいても、例えば優先順位に基づいて動的にオブジェクトをプールすることが可能となり、オブジェクトをプールする際にルーティングできず実行時でないとは明確にならないオブジェクトに対しても、有効なオブジェクトプールを提供することができる。

【0038】

次に、本実施の形態におけるクライアント 11 からのオブジェクト生成要求時の動作について、図 16 のフローチャートを用いて説明する。

まず、オブジェクトプール 30 のクライアント要求解析部 41 は、クライアント 11 からのリクエストの内容を解析し、オブジェクト生成処理部 42 へ処理を依頼する(ブロック 161)。この依頼を受けたオブジェクト生成処理部 42 は、生成を要求されたオブジェクトのサーバー名、プロジェクト名、クラス名をオブジェクト管理部 32 に転送し、オブジェクトの取得要求を行う(ブロック 162)。この要求を受けたオブジェクト管理部 32 は、空きオブジェクト管理部 34 に対して、該当するクラスのオブジェクトの取得要求を行う(ブロック 163)。この取得要求を受けた空きオブジェクト管理部 34 は、空きオブジェクト管理テーブル内に要求されたクラスのオブジェクトが存在する場合、そのオブジェクトの実行状態をチェックした上で、使用可能な状態であればそのオブジェクトをオブジェクト管理部 32 に渡し、空きオブジェクト管理テーブルに存在していたそのオブジェクトの項目を削除する。空きオブジェクト管理テーブル内に要求されたオブジェクトが存在している場合でも、オブジェクトが使用できない状態にあるときは、それをオブジェクト管理部 32 に通知する(ブロック 164)。オブジェクト管理部 32 は、空きオブジェクト管理部 34 から該当するオブジェクトを取得できなかった場合、新たにオブジェクトを生成する。活動中オブジェクト管理部 33 に対して、空きオブジェクト管理部 34 から取得したオブジェクト、あるいは新たに生成したオブジェクトの情報を、活動中オブジェクト管理テーブルに追加するよう依頼し、オブジェクト生成処理部 42 に要求されたオブジェクトを返す。空きオブジェクト管理部 34 からオブジェクトが使用不可能な状態にあると通知された場合には、新たなオブジェクトの生成は行わず、その由をオブジェ

クト生成処理部 42 に返し(ブロック 165)、一連の動作が終了する。

【0039】

次に、クライアント 11 からのオブジェクト削除要求時の動作について、図 17 のフローチャートを用いて説明する。まず、オブジェクトプール 30 のクライアント要求解析部 41 は、クライアント 11 からのリクエストの内容を解析し、オブジェクト削除処理部 44 へ処理を依頼する(ブロック 171)。この依頼を受けたオブジェクト削除処理部 44 は、削除を要求されたオブジェクトをオブジェクト管理部 32 に転送し、オブジェクトの削除要求を行う(ブロック 172)。この要求を受けたオブジェクト管理部 32 は、活動中オブジェクト管理部 33 に対して、要求されたオブジェクトを活動中オブジェクト管理テーブルから削除するように依頼すると共に、空きオブジェクト管理部 34 に対して、そのオブジェクトの空きオブジェクト管理テーブルへの追加を要求する(ブロック 173)。この要求を受けた空きオブジェクト管理部 34 は、オブジェクト管理部 32 からの依頼に基づき、該当オブジェクトの情報を空きオブジェクト管理テーブルへ追加する。続いて、空きオブジェクトの数と予め定められた最大プール数(図 8)を比較し、空きオブジェクトの数が最大プール数を超えていた場合、優先順位の一番低いオブジェクトをプールから外し、オブジェクト管理部 32 に対して、そのオブジェクトの削除を要求する(ブロック 174)。オブジェクト管理部 32 は、空きオブジェクト管理部 34 からのオブジェクトの削除要求に従い、オブジェクトを削除し(ブロック 175)、削除要求時の動作が終了する。

【0040】

次に、クライアント 11 からのメソッドコール処理要求があった時の動作について、図 18 のフローチャートを用いて説明する。

まず、オブジェクトプール 30 のクライアント要求解析部 41 は、クライアント 11 からのリクエストの内容を解析し、メソッドコール処理部 43 へ処理を依頼する(ブロック 181)。この依頼を受けたメソッドコール処理部 43 は、オブジェクト管理部 32 に対して、処理要求のあったオブジェクトを請求する(ブロック 182)。処理要求を受けたオブジェクト管理部 32 は、要求されたオブジェクトを活動中オブジェクト管理部 33 から取得し、メソッドコール処理部 43

に引き渡す。この時、活動中オブジェクト管理部 33 は、活動中オブジェクト管理テーブルをチェックし、その実行状態も合わせてメソッドコール処理部 43 に戻す(ブロック 183)。メソッドコール処理部 43 は、オブジェクト管理部 32 から受け取ったオブジェクトの実行状態をチェックし、可能であれば取得したオブジェクトにメソッドを実行させ、結果をクライアント 11 に戻す。オブジェクトが実行可能状態にない場合には、クライアント 11 に対して即座にエラーを通知し(ブロック 184)、一連のメソッドコール処理要求時の動作が終了する。

【0041】

尚、本実施の形態におけるオブジェクトプールの終了動作時(図 15)にて、空きオブジェクト管理テーブル(図 7)で実行状態が停止中であり、更に再度起動することがないプロジェクトに関するオブジェクトを消去するか否かは、その実装の形態によって判断される。但し、プロジェクト完全終了時に空きオブジェクト管理テーブルからそのオブジェクトを消去した場合には、クライアント 11 からリクエストがあると、アプリケーションを実行するアプリケーションサーバー 20 に対してオブジェクト作成を要求し、アプリケーションサーバー 20 からのエラー返答を待ってからクライアント 11 に対してエラーを返すことになる。一方、そのようなオブジェクトを消去しなかった場合には、アプリケーションサーバー 20 からの返答を待たずにクライアント 11 に対してエラーを返すことが可能となり、より迅速な対応が可能となる点で優れている。

【0042】

次に、本実施の形態における優先順位の更新動作について、図 19 及び図 20 を用いて説明する。

図 19(a)、(b)はオブジェクト削除要求処理前の各テーブルエントリーの例を示し、図 20(a)、(b)はオブジェクト削除要求処理後の各テーブルエントリーの例を示している。まず、クライアント 11 からのオブジェクト削除要求時の動作について説明する。図 19(a)、(b)に示すように、例えばある時点で、活動中オブジェクト管理テーブルにはクラス名 X であるオブジェクト X が、空きオブジェクト管理テーブルにはクラス名 A、B であるオブジェクト A、B が存在していたものと仮定する。クライアント 11 がオブジェクト X の削除を要求すると

、図5に示したオブジェクト削除処理部44はオブジェクト管理部32に対してその処理を依頼する。この依頼を受けたオブジェクト管理部32は、活動中オブジェクト管理部33および空きオブジェクト管理部34に命じて、活動中オブジェクト管理テーブルにあったオブジェクトX（図19(a))を空きオブジェクト管理テーブルへ移す。このとき、優先順位の算出方法としてそのオブジェクトへの最終アクセス時間を採用していた場合、空きオブジェクト管理部34はオブジェクトXに最も高い優先順位を割り振り、オブジェクトA、Bの優先順位はそれぞれ下げられる。その結果、活動中オブジェクト管理テーブルは図20(a)に示すものとなり、空きオブジェクト管理テーブルは図20(b)に示すような内容となる。

【0043】

次に、オブジェクトプール30でのイベント受信時の優先順位決定動作について、例を挙げて説明する。オブジェクトプール30のオブジェクト管理部32は、イベント処理部31から通知されたイベントに従い、活動中オブジェクト管理部33と空きオブジェクト管理部34に対して、それぞれが管理するテーブルの更新を依頼する。このとき、空きオブジェクト管理部34は、実行順位と共に優先順位の変更を行う。一方、活動中オブジェクト管理テーブルには優先順位の情報保持されていないので、活動中オブジェクト管理部33は、オブジェクトのアプリケーション実行環境(アプリケーションサーバー20)での実行状態のみを更新する。

優先順位の更新としては、例えば、プロジェクトの停止、一時停止、ビジー状態などのイベントを受信した場合に、そのオブジェクトの優先順位を下げるものが挙げられる。また、例えば、プロジェクトの開始処理完了、実行準備完了などのイベントを受信した場合に、そのオブジェクトの優先順位を上げるように構成することも可能である。本実施の形態によれば、実装の形態によってここに掲げた以外にも様々な内容のイベントに対して優先順位をダイナミックに更新することが可能であり、オブジェクトプール30の立ち上げである起動時にクライアント11が使用するであろうオブジェクトを有効に決定することで、効果的なオブジェクトプールを構築することが可能となる。

【 0 0 4 4 】

図 2 1 は、本実施の形態を用いたアプリケーションシステムの一例を示す図である。同図において、アプリケーションサーバー 7 1 は、受注・発注処理や、顧客情報、人事管理情報等のアプリケーションを実行するアプリケーション実行環境である。また、サーバー 7 2 は、本実施の形態におけるオブジェクトプールとして機能しており、アプリケーションサーバー 7 1 とは一般的に LAN 等の通信ネットワーク 7 4 を介して接続されている。また、クライアント 7 3 は、例えば希望する作業をメソッドという形で要求してサーバーに対して処理を依頼し、その結果を受け取るもので、WAN や LAN 等からなる通信ネットワーク 7 5 を介してサーバー 7 2 に接続されている。

【 0 0 4 5 】

今、クライアント 7 3 の 1 つが Web を経由して文房具などのオフィス製品を注文する場合を例にとる。図 2 2 は、このクライアントアプリケーションの画面イメージを示した図である。このアプリケーションは、商品を Web を経由して注文を行うことを可能にする一般的な受発注処理プログラムである。まず、クライアント 7 3 は、クリエイトオブジェクト、即ち、どのサーバー上の(受注処理業務の)受注クラスのオブジェクトを作って欲しい旨を指示する。このクライアントは、例えばアプリケーションサーバーが提供する Applet を使用してサーバーとコミュニケーションを行うことが可能である。図 2 2 のディスプレイ 8 1 の Web ページ上に存在するオーダーボタン 8 2 は、HTML 及び Java Script で所定の定義がなされおり、ユーザーがオーダーボタン 8 2 を押下すると、所定の関数が呼ばれ、サーバー 7 2 を経由し、リモートサーバーである受注・発注処理を実行するアプリケーションサーバー 7 1 に定義されている“Order”メソッドが実行される。クライアント 7 3 は、オブジェクトプールであるサーバー 7 2 をパススルーし、受注オブジェクトに対して価格や在庫の照会、買いやキャンセル等をメソッドとしてオブジェクトに投げかけるように構成されている。尚、本実施の形態では、サーバー 7 2 でプールされるプロキシオブジェクトを介してオブジェクトを使用する形態にあり、受注・発注処理のプロジェクトを実行するオブジェクトの状態変化をサーバー 7 2 が認識していることから、例えば異常終

了があった場合等にクライアント 73 に対して有効なオブジェクトでない旨をいち早く通知でき、冗長な処理を省略することが可能となる。

【0046】

[実施の形態 2]

実施の形態 1 では、オブジェクトプールがプロキシオブジェクトの形態でプールし、アプリケーション実行環境は別に備えていたが、本実施の形態では、自らアプリケーション実行環境であるサーバーがオブジェクトプールとして機能するものである。

尚、実施の形態 1 と同様の構成については同様の符号を付し、ここではその詳細な説明を省略する。

【0047】

図 23 は、実施の形態 2 におけるクライアントサーバーシステムの全体構成を説明するための図である。同図において、オブジェクトプール 90 は、アプリケーション実行環境としてオブジェクト 91 を自ら生成すると共に、クライアント 11 に対して効果的なオブジェクトプールとして機能している。

図 24 は、オブジェクトプール 90 の機能ブロックを説明するための図である。実施の形態 1 におけるオブジェクトプール 30 (図 5) とは、イベント処理部 31 を備えていない点で構成上の異なりがある。

【0048】

次に、図 24 を用いて各処理の流れについて説明する。

まず、クライアント 11 からオブジェクト生成要求があった場合、クライアント要求解析部 41 は、リクエストのタイプを解析し、オブジェクト生成処理部 42 へ処理を依頼する。この依頼を受けたオブジェクト生成処理部 42 は、オブジェクト管理部 32 にオブジェクトの取得を依頼する。オブジェクト管理部 32 では、空きオブジェクト管理部 34 に問い合わせ、該当するクラスのオブジェクトがあればそれを渡し、なければオブジェクトを生成してそれを渡す。

また、クライアント 11 からオブジェクト削除要求があった場合、クライアント要求解析部 41 は、リクエストのタイプを解析し、オブジェクト削除処理部 44 へ処理を依頼する。この依頼を受けたオブジェクト削除処理部 44 は、オブジ

ェクト管理部 32 にオブジェクトの削除を依頼する。オブジェクト管理部 32 では、空きオブジェクトとして、空きオブジェクト管理部 34 にそのオブジェクトを登録するように要求する。このとき、空きオブジェクトの数が最大プール数に達していたら、優先順位が一番低いオブジェクトをプールから外し、実際に削除する。この優先順位はオブジェクトの最終アクセス時間等で決定することができる。

【0049】

次に、オブジェクトプール 90 であるサーバプロセスを終了する場合には、終了処理部 37 は、空きオブジェクト管理部 34 からオブジェクトのタイプと優先順位のリストを取得し、オブジェクト情報記憶部 35 に保存する。一方、サーバプロセスを開始するイニシャル時には、開始処理部 36 は、終了時に保存したデータから優先順位の高い順に、初期プール数に達するまでオブジェクトの生成要求をオブジェクト管理部 32 に対して行う。この結果、アプリケーション実行環境にあるサーバには、分散オブジェクトを利用したアプリケーション下にて、クライアントが使用するであろうオブジェクトをイニシャル時に事前に生成することが可能となり、有効なオブジェクトプールとして機能することができる。特に、例えば、最終アクセスのあったオブジェクトの優先順位を高くした状態にてサーバプロセスが終了され、イニシャル時にその優先順位に基づいて最終アクセスのあったオブジェクトからオブジェクトの生成がなされるように構成することで、再度使用される可能性が高い最終アクセスのあったオブジェクトをまず生成することが可能となる等、オブジェクトプールの有用性を向上させることができる。

【0050】

尚、これらの実施形態を実行するためのソフトウェアは、CD-ROM等の記憶媒体を介して各サーバにインストールすることで実行させることができる。そのインストールは、実施の形態 1 では、オブジェクトプール 30 として機能するサーバへのインストールと、アプリケーションサーバ 20 へのインストールの態様が挙げられ、実施の形態 2 ではオブジェクトプール 90 として機能するサーバへのインストールの態様が挙げられる。また、予めハードディスク等の

記憶媒体にインストールされている場合の他、CD-ROM等の記憶媒体を介さずにネット等を介してソフトウェアをダウンロードすることによっても達成できる。その際には、プログラムを記憶する記憶手段と、そのプログラムを読み出して送信する送信手段をPC等に具備させ、インターネット等に接続されたプログラムを実行する他のPCからの要求によってそのPCにダウンロードさせるように構成すれば良い。また、各クライアントに対しても同様なプログラムインストール方法をもって、これらの実施形態におけるクライアントサーバシステムを形成することが可能である。

【0051】

以上説明したように、実施の形態1及び2によれば、オブジェクトプールの立ち上げ時に動的にオブジェクトをプールすることが可能となり、分散オブジェクトを利用したアプリケーションにおけるオブジェクトの作成時に費やされる処理時間を有効に短縮することができる。また、本実施の形態によれば、クライアントから接続されているサーバーのバックエンドに幾つかのアプリケーションサーバーがあり、そのアプリケーションサーバーにて実オブジェクトを作成するようなモデルであっても、そのサーバーで実行されているアプリケーションの状態変化を適切に把握することが可能となり、常に有効なオブジェクトをプールしておくことが可能となる。

【0052】

更に、本実施の形態による方式では、アプリケーション実行環境とオブジェクトプール内の管理オブジェクトが、直接、イベントの送受信を実行するため、ユーザーがプール制御用のプログラムを用意する必要がなく、更にはアプリケーションの起動時などユーザーオブジェクトが存在しない時点でのオブジェクトプールの制御を行うことも可能である。またこのとき、サーバーとオブジェクトプールの間で発行されるイベントを、CORBA(Common ORB Architecture)のイベントサービスなど、分散環境に対応したORB(Object Request Broker)を利用することにより、容易にネットワークを介してイベント通知を行うことが可能となる。このCORBAのように公開された標準的な形式でイベントを定義することで、異なるベンダー間での相互接続性も期待できる。

【 0 0 5 3 】

【発明の効果】

以上説明したように、本発明によれば、サーバーの起動後、クライアントから最初にオブジェクトの生成要求があったときの応答時間を短縮することができる。また、アプリケーションの状態を把握することでオブジェクトプールの有用性を向上させることができる。

【図面の簡単な説明】

【図 1】 本発明の形態におけるクライアントサーバーシステムの全体構成を説明するための図である。

【図 2】 アプリケーション実行環境の機能ブロックを説明するための図である。

【図 3】 イベント生成部 2 1 において生成されるイベント通知管理テーブルを示すものである。

【図 4】 プロジェクト実行監視部 2 5 において更新されるプロジェクト実行監視テーブルを示すものである。

【図 5】 オブジェクトプール 3 0 の機能ブロックを説明するための図である。

【図 6】 活動中オブジェクト管理部 3 3 において用いられる活動中オブジェクト管理テーブルを示すものである。

【図 7】 空きオブジェクト管理部 3 4 にて用いられる空きオブジェクト管理テーブルを示すものである。

【図 8】 本実施の形態におけるオブジェクトプールプロパティ管理テーブルを示すものである。

【図 9】 オブジェクト情報記憶部 3 5 に記憶されるプールオブジェクト情報を示すものである。

【図 1 0】 プロジェクトの開始動作を説明するためのフローチャートである。

【図 1 1】 プロジェクトの停止動作を説明するためのフローチャートである。

【図 1 2】 プロジェクトの異常終了処理動作を説明するためのフローチャートである。

【図 1 3】 イベント受信時の動作を説明するためのフローチャートである。

【図 1 4】 オブジェクトプールの開始動作を説明するためのフローチャートである。

【図 1 5】 オブジェクトプールの終了動作説明するためのフローチャートである。

【図 1 6】 クライアントからのオブジェクト生成要求時の動作について説明するためのフローチャートである。

【図 1 7】 クライアントからのオブジェクト削除要求時の動作について説明するためのフローチャートである。

【図 1 8】 クライアントからのメソッドコール処理要求があった時の動作について説明するためのフローチャートである。

【図 1 9】 (a)、(b)はオブジェクト削除要求処理前の各テーブルエントリーの例を示すものである。

【図 2 0】 (a)、(b)はオブジェクト削除要求処理後の各テーブルエントリーの例を示すものである。

【図 2 1】 本実施の形態を用いたアプリケーションシステムの一例を示す図である。

【図 2 2】 クライアントアプリケーションの画面イメージを示した図である。

【図 2 3】 実施の形態 2 におけるクライアントサーバーシステムの全体構成を説明するための図である。

【図 2 4】 オブジェクトプールの機能ブロックを説明するための図である。

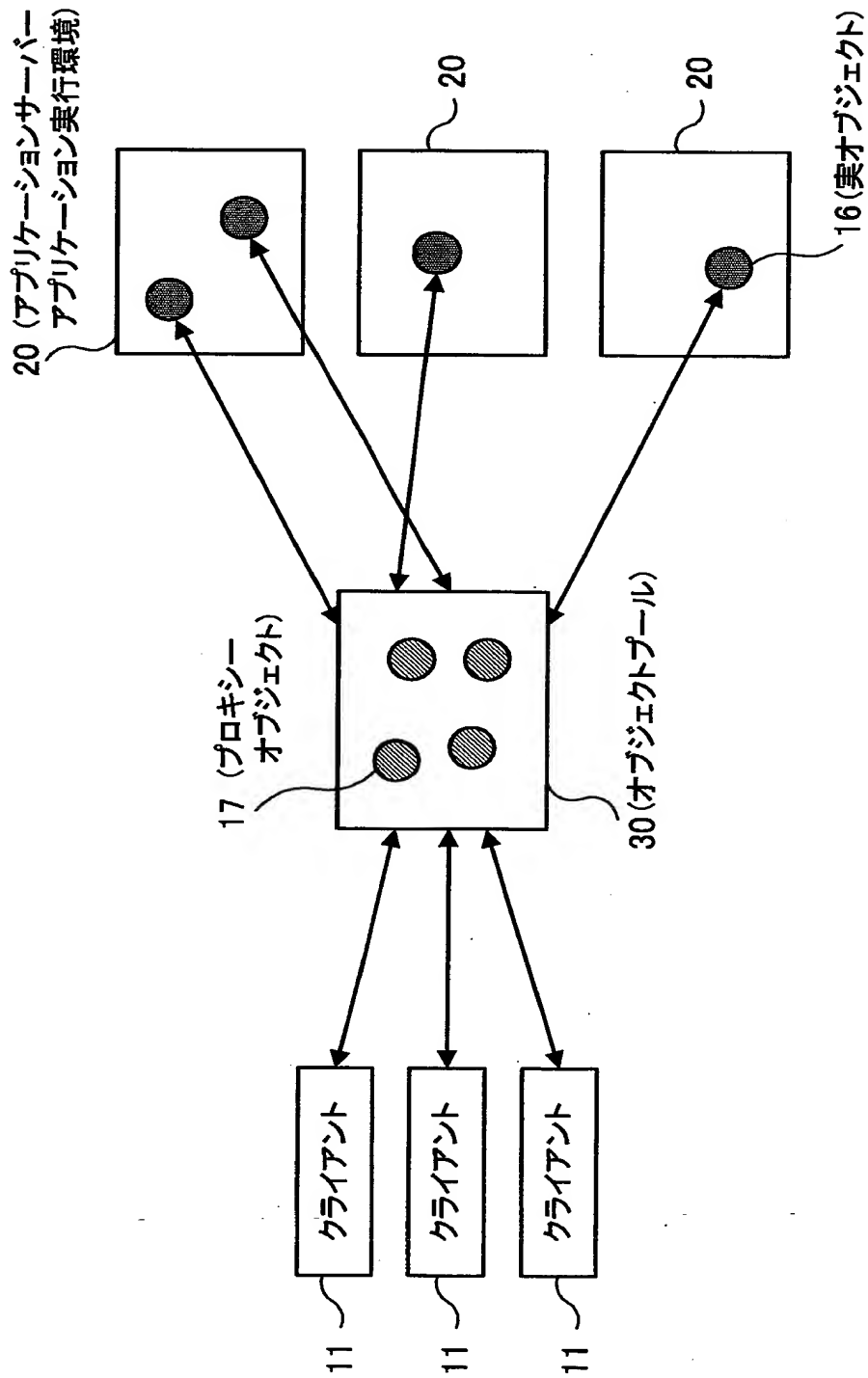
【符号の説明】

1 1 …クライアント、1 6 …実オブジェクト、1 7 …プロキシオブジェクト、
2 0 …アプリケーションサーバー/アプリケーション実行環境、2 1 …イベント

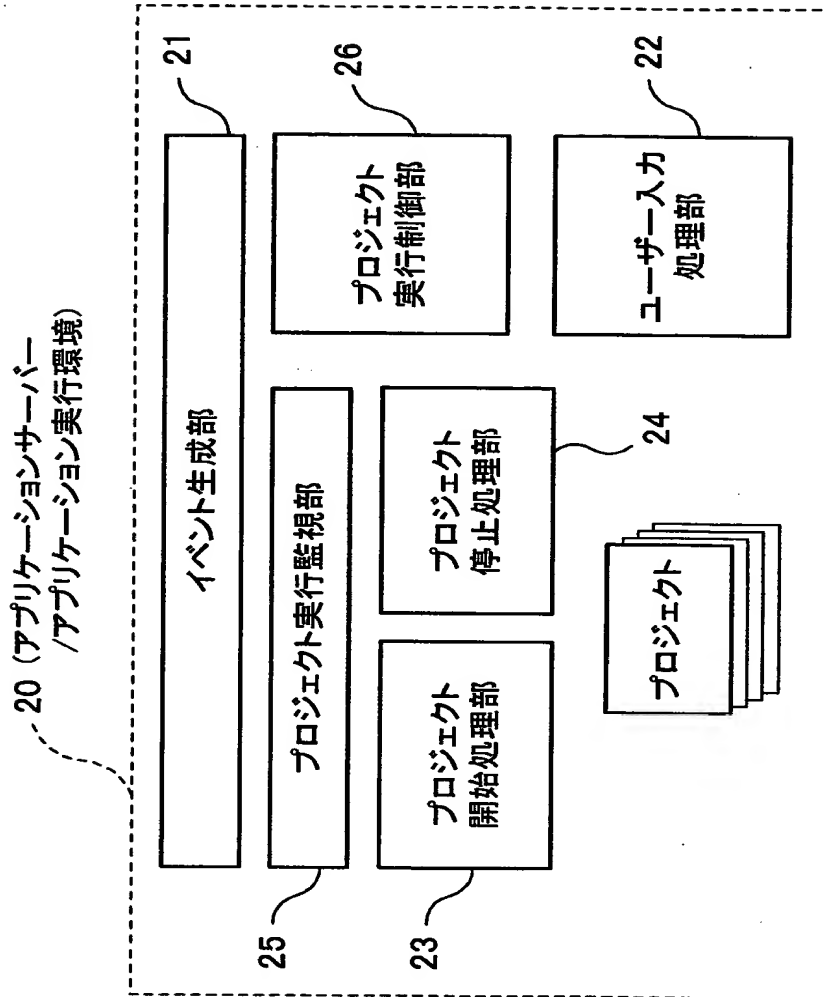
生成部、22…ユーザー入力処理部、23…プロジェクト開始処理部、24…プロジェクト停止処理部、25…プロジェクト実行監視部、26…プロジェクト実行制御部、30…オブジェクトプール、31…イベント処理部、32…オブジェクト管理部、33…活動中オブジェクト管理部、34…空きオブジェクト管理部、35…オブジェクト情報記憶部、36…開始処理部、37…終了処理部、41…クライアント要求解析部、42…オブジェクト管理部、43…メソッドコール処理部、44…オブジェクト削除処理部、71…アプリケーションサーバー、72…サーバー、73…クライアント、74、75…通信ネットワーク、81…ディスプレイ、82…オーダーボタン、90…オブジェクトプール、91…オブジェクト

【書類名】 図面

【図 1】



【図 2】



【図 3】

イベント通知管理テーブル

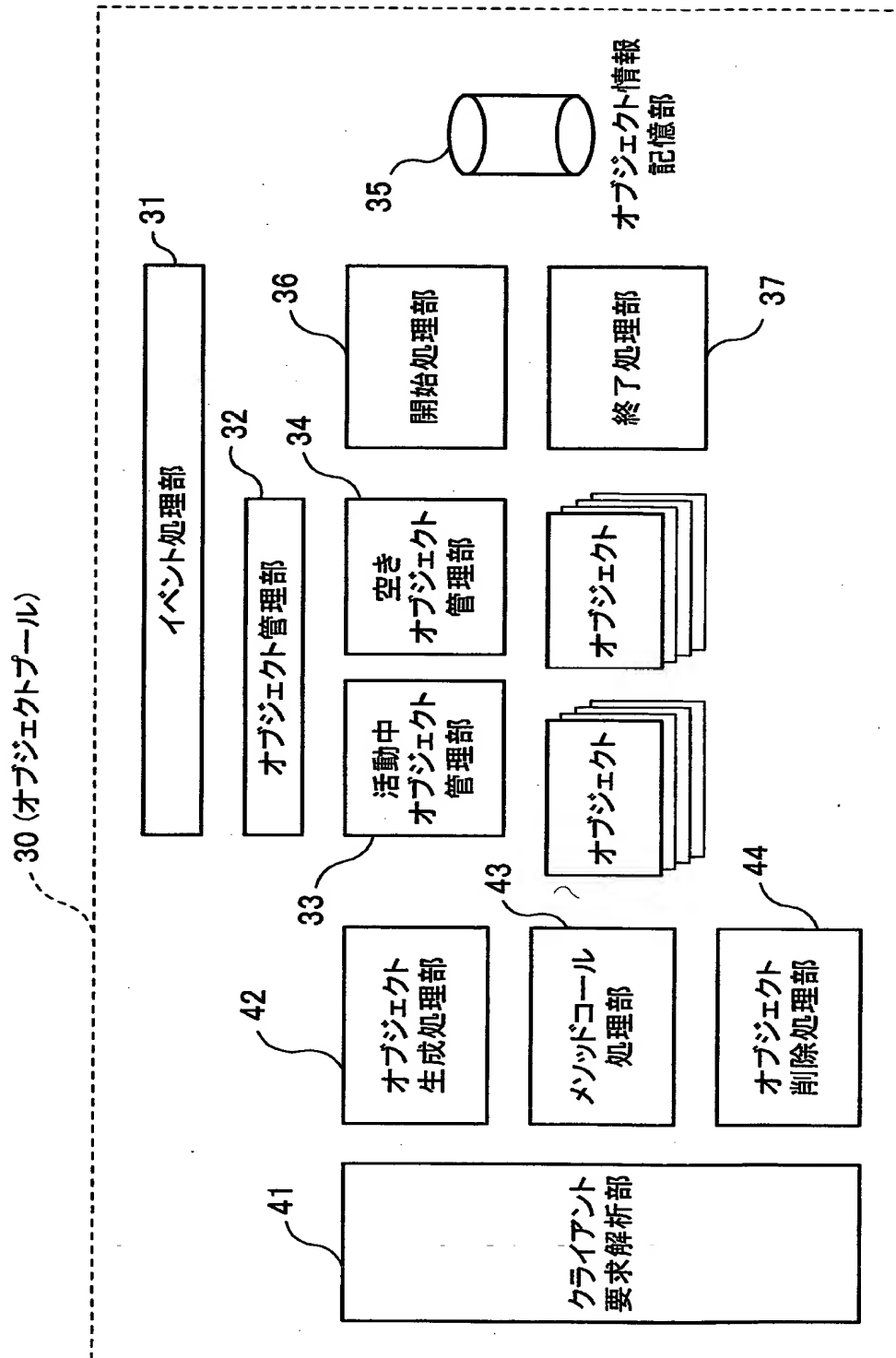
オブジェクトプール ID	ホスト名	イベントフィルタ
112233	Host A	All
445566	Host B	Only Error
778899	Host C	Warning

【図 4】

プロジェクト実行監視テーブル

プロジェクト ID	プロジェクト名	イベントフィルタ
112233	プロジェクト A	開始
445566	プロジェクト B	停止中
778899	プロジェクト C	実行中

【図 5】



【図 6】

活動中オブジェクト管理テーブル

オブジェクトID	クラス名	サーバー名	プロジェクト名	実行状態
112233	クラス A	サーバー 3	プロジェクト X	実行中
445566	クラス E	サーバー 5	プロジェクト Y	停止中
778899	クラス C	サーバー 1	プロジェクト Z	実行中

【図 7】

空きオブジェクト管理テーブル

優先順位	オブジェクトID	クラス名	サーバー名	プロジェクト名	実行状態
1	112233	クラス A	サーバー 3	プロジェクト X	実行中
2	445566	クラス E	サーバー 5	プロジェクト Y	停止中
3	778899	クラス C	サーバー 1	プロジェクト Z	実行中

【図 8】

オブジェクトプールプロパティ管理テーブル

プロパティ	値
最大プール数	500
初期プール数	100

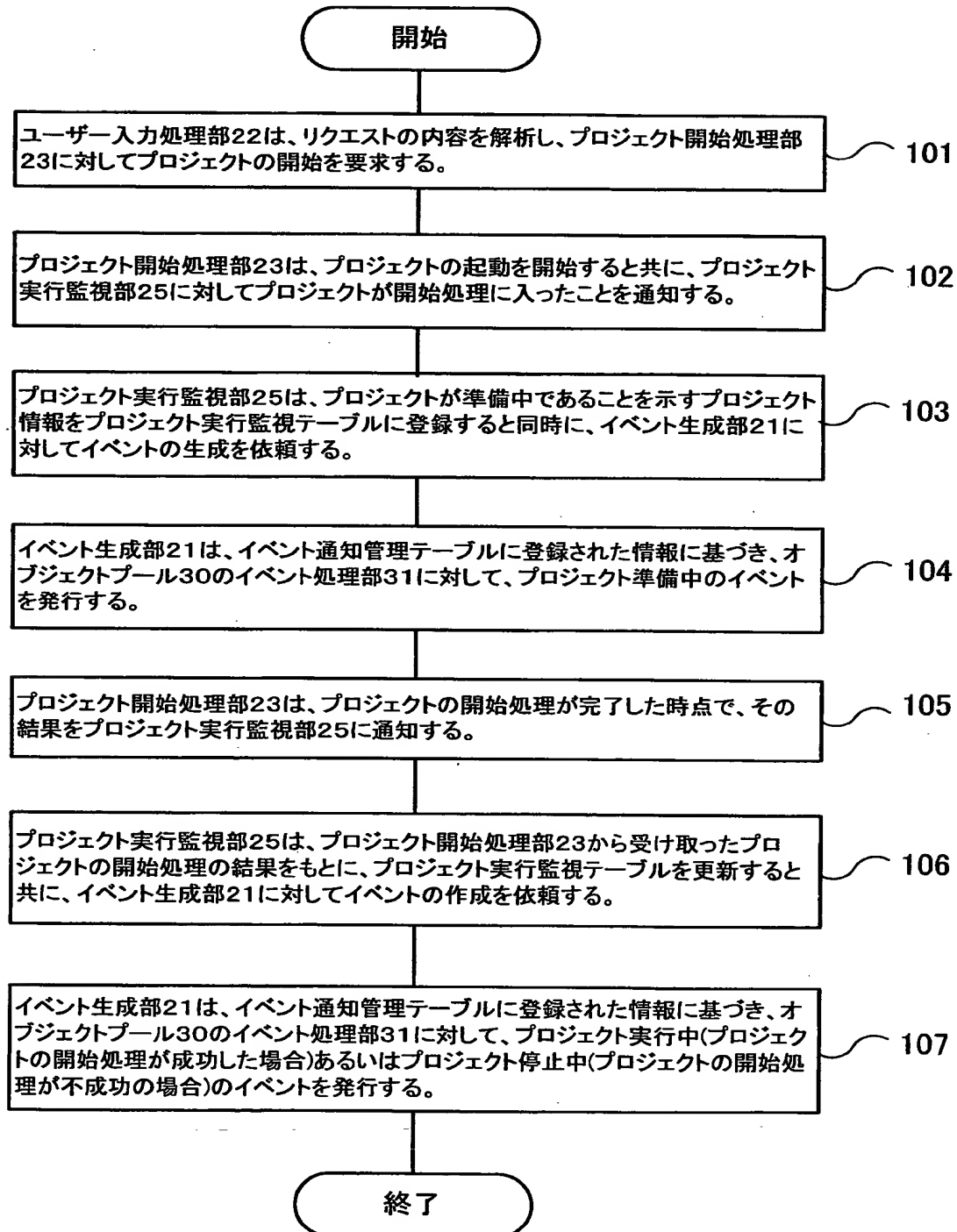
【図 9】

プールオブジェクト情報

優先順位	クラス名	サーバー名	プロジェクト名
1	クラス A	サーバー 3	プロジェクトX
2	クラス E	サーバー 5	プロジェクトY
3	クラス C	サーバー 1	プロジェクトZ

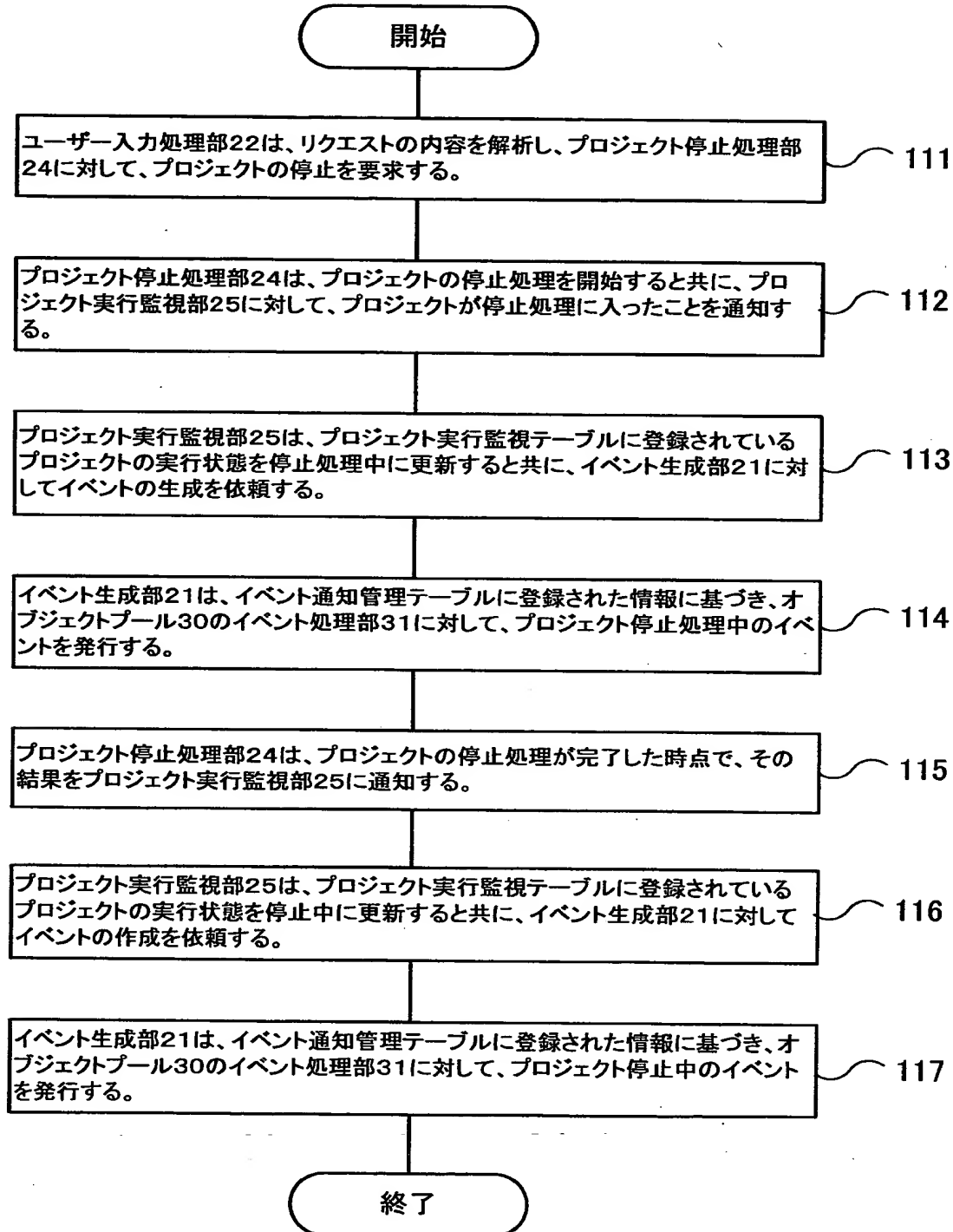
【図 10】

プロジェクトの開始動作



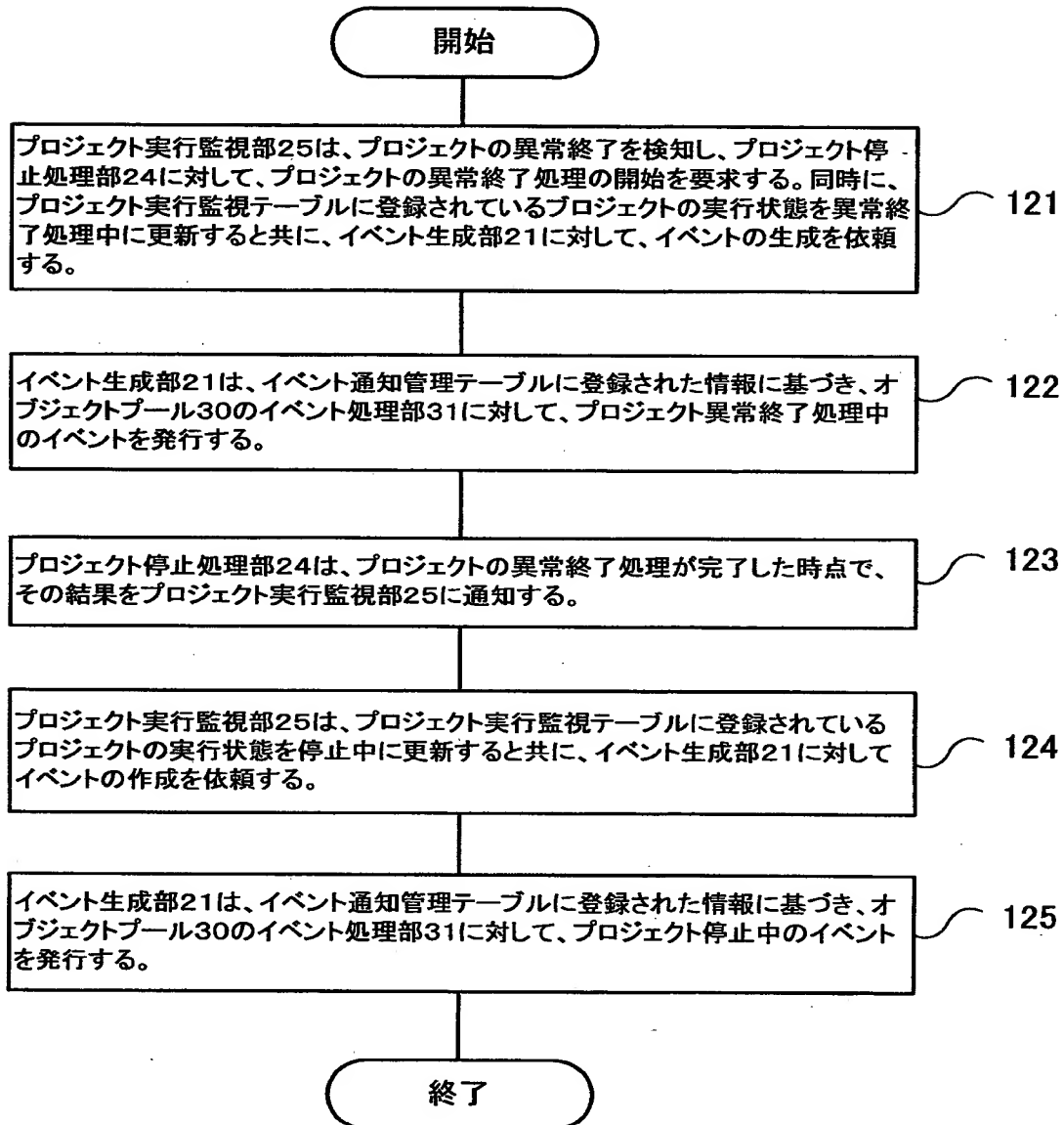
【図 11】

プロジェクトの停止動作



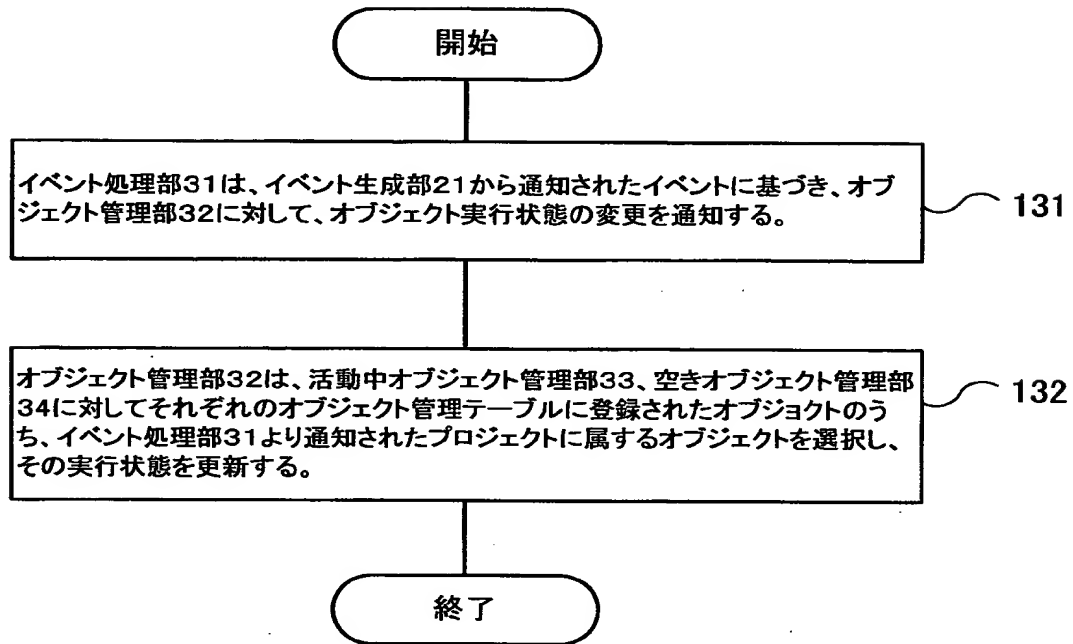
【図 1 2】

プロジェクトの異常終了処理動作



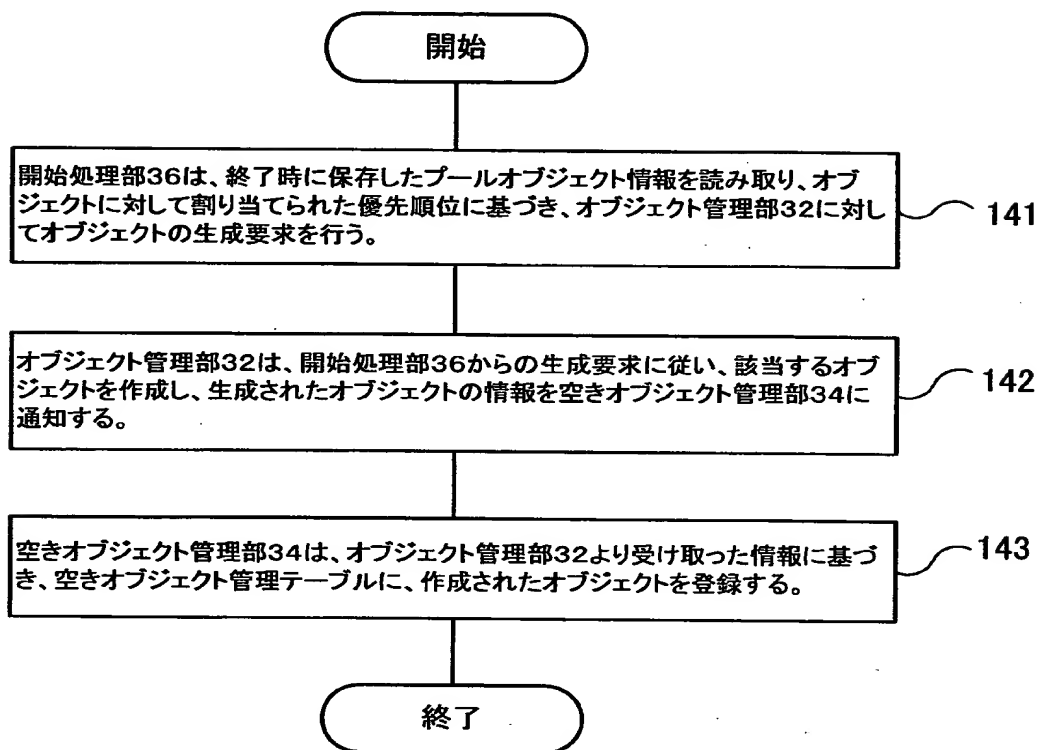
【図 1 3】

オブジェクトプールでのイベント受信時の動作



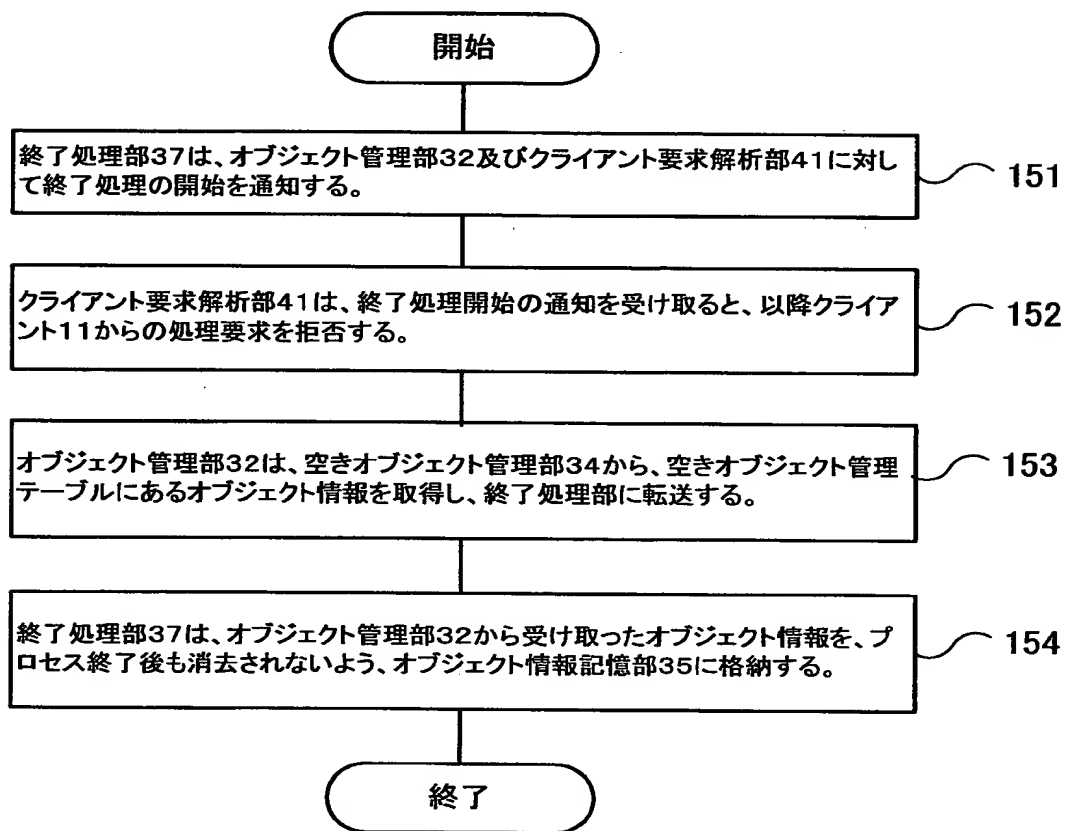
【図 1 4】

オブジェクトプールの開始動作



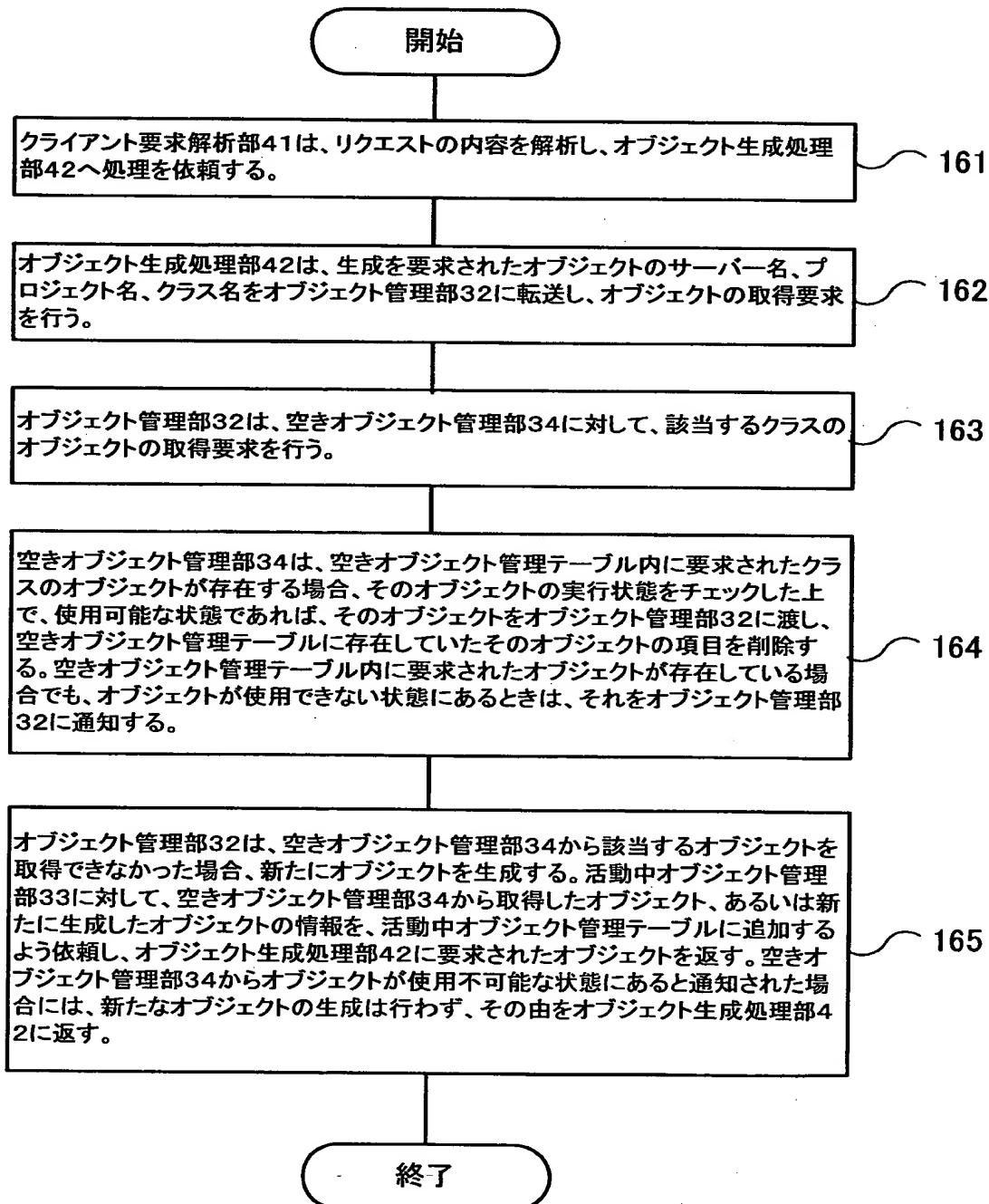
【図 1 5】

オブジェクトプールの終了動作



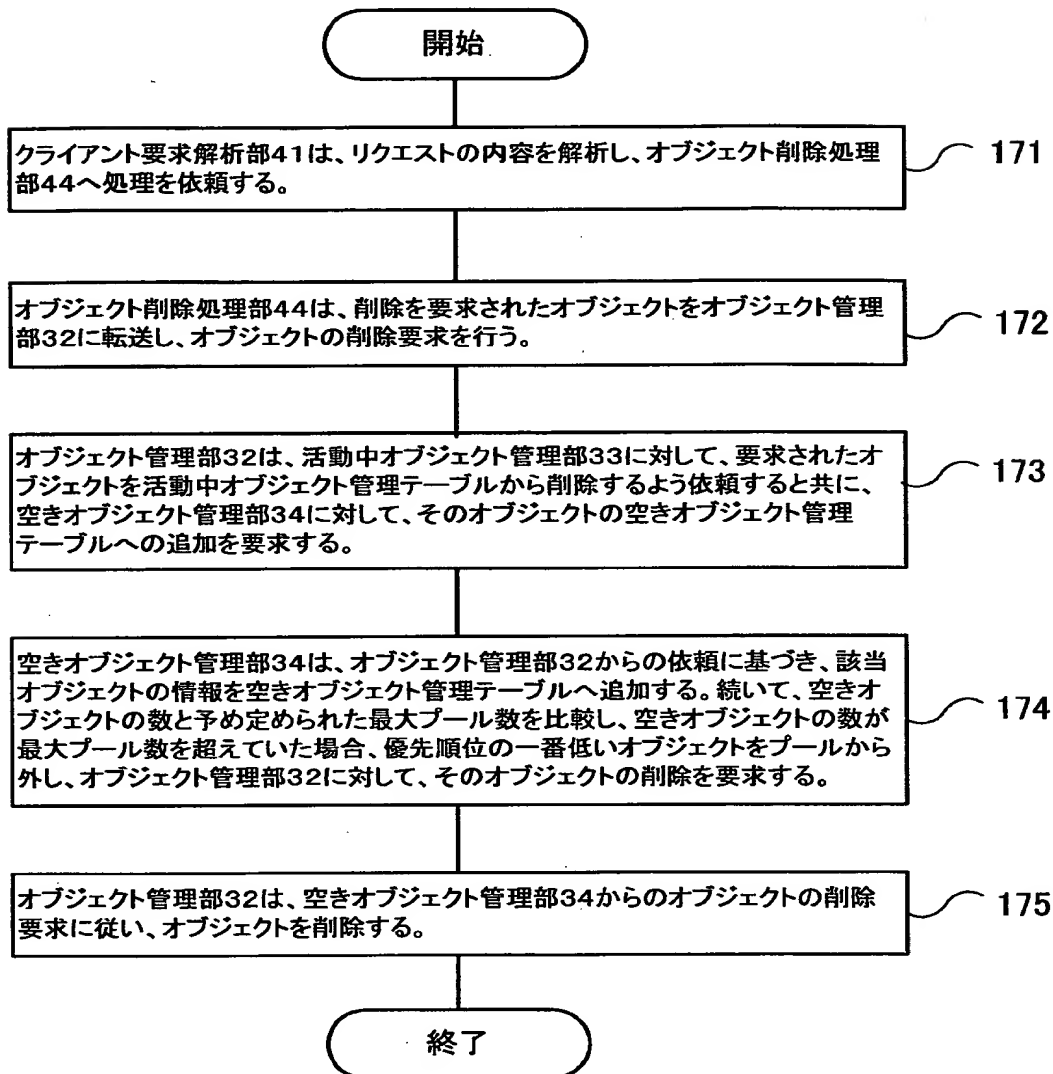
【図 16】

クライアントからのオブジェクト生成要求時の動作



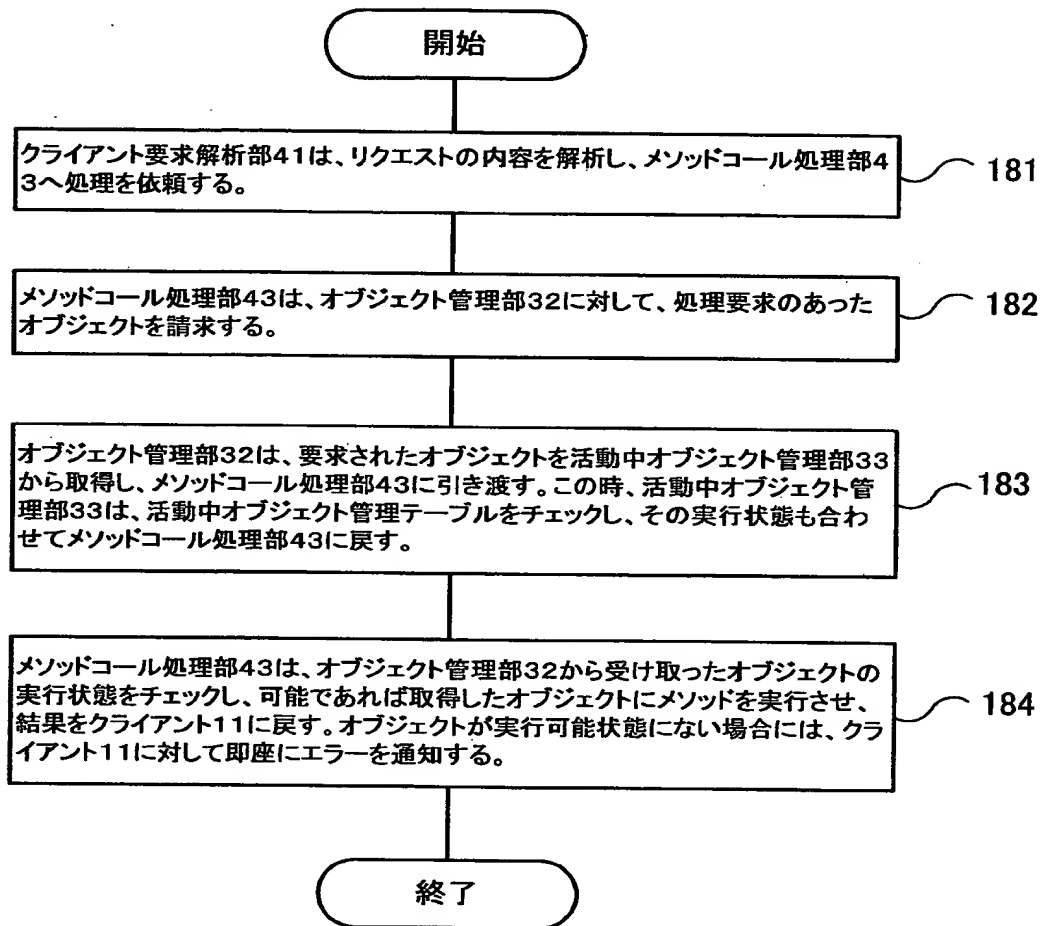
【図 17】

クライアントからのオブジェクト削除要求時の動作



【図 18】

クライアントからのメソッドコール処理要求時の動作



【図 19】

活動中オブジェクト管理テーブル

オブジェクトID	クラス名	...
112233	クラス X	...
.	.	.
.	.	.

(a)

空きオブジェクト管理テーブル

オブジェクトID	優先順位	クラス名	...	最終アクセス時間
223344	1	クラス A	...	08:08:08
334455	2	クラス B	...	07:07:07
.
.

(b)

【図 20】

活動中オブジェクト管理テーブル

オブジェクトID	クラス名	...
.	.	.
.	.	.

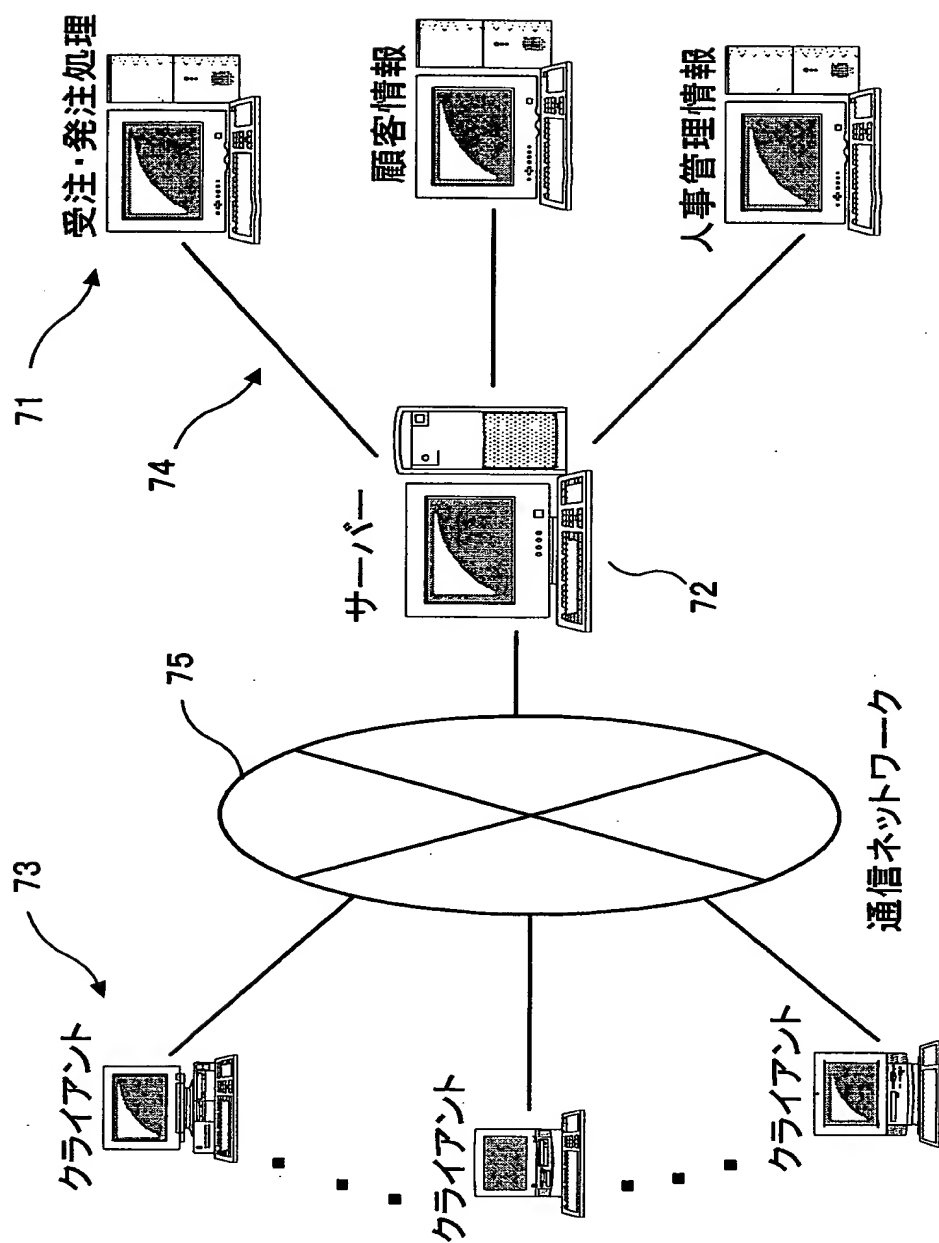
(a)

空きオブジェクト管理テーブル

オブジェクトID	優先順位	クラス名	...	最終アクセス時間
112233	1	クラス X	...	09:09:09
223344	2	クラス A	...	08:08:08
334455	3	クラス B	...	07:07:07

(b)

【図 21】



【図 2 2】

81

Order Form

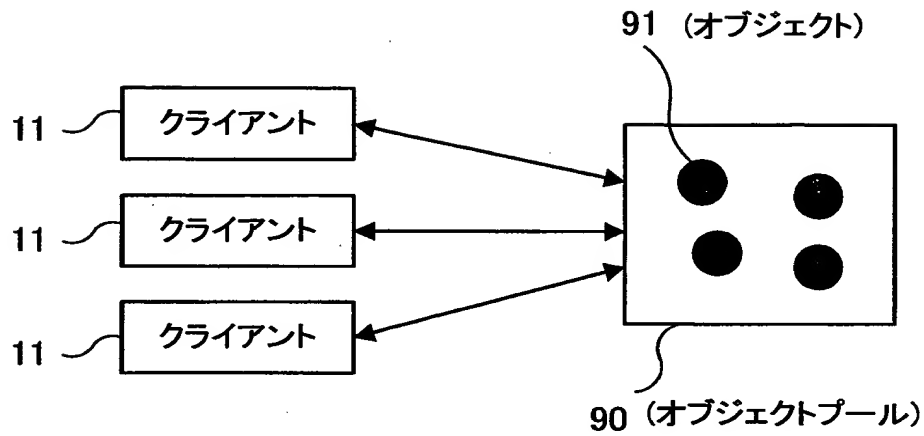
Staff Name : Taro Yamada
Office Name : New York (Code : NO1)
Created Date : 08/30/1999 15:44:41

82

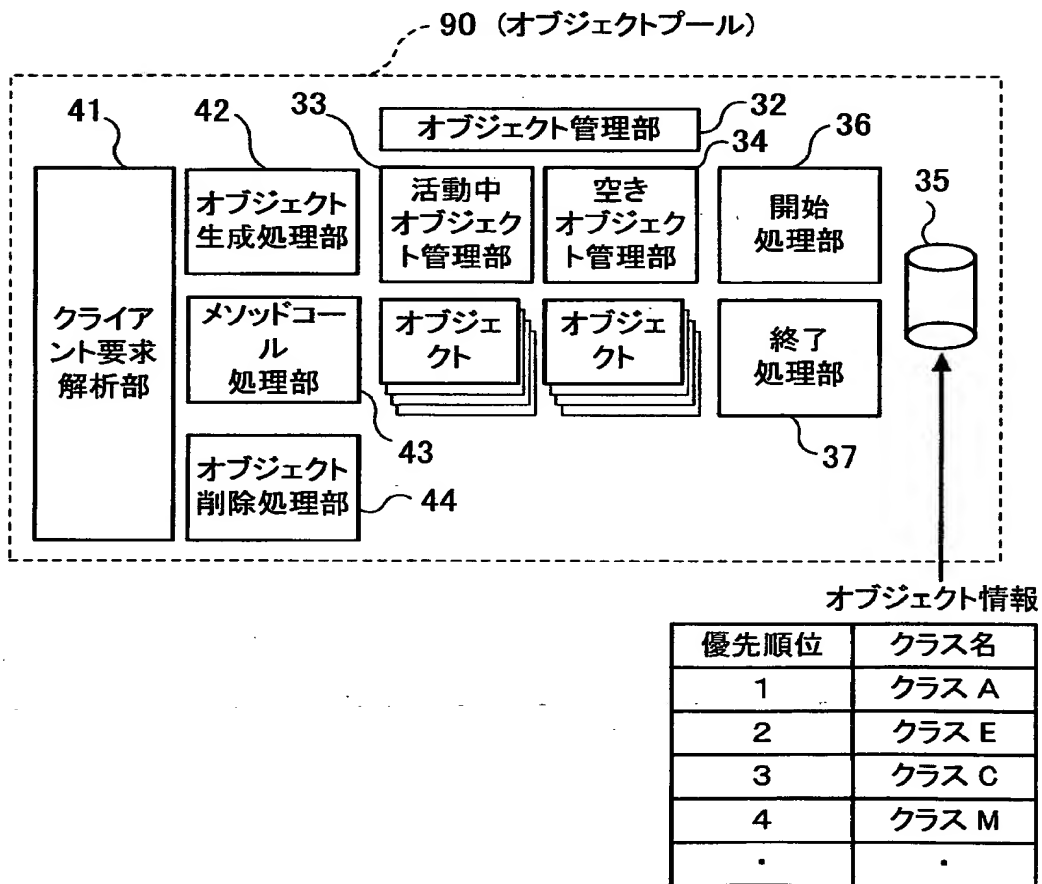
Validate Order Clear

Customer Name	<input type="text"/>	Customer Code	<input type="text"/>
Category	<input type="text"/>		
Product Name	<input type="text"/>	Product Code	<input type="text"/>
Quantity	<input type="text"/>	Requested Ship Date	<input type="text"/>
Discount Ratio	<input type="text"/>	Final Ship Date	<input type="text"/>
Amount	<input type="text"/>	Order Number	<input type="text"/>

【図 2 3】



【図 2 4】



【書類名】 要約書

【要約】

【課題】 サーバーの起動後、クライアントから最初にオブジェクトの生成要求があったときの応答時間を短縮し、オブジェクトプールの有用性を向上させる。

【解決手段】 通信ネットワークに接続され、オブジェクトに対するアクセス要求を行うクライアント 1 1 と、クライアント 1 1 の要求に基づいて実オブジェクト 1 6 によりアプリケーションを実行するアプリケーションサーバー 2 0 と、クライアント 1 1 及びアプリケーションサーバー 2 0 と接続され、実オブジェクト 1 6 に対応するプロキシオブジェクト 1 7 をプーリングすると共に、実オブジェクト管理情報を保持するオブジェクトプール 3 0 とを備え、アプリケーションサーバー 2 0 はアプリケーションの状態変化に応じてオブジェクトプール 3 0 に対してイベントを通知すると共に、オブジェクトプール 3 0 はアプリケーションサーバー 2 0 からのイベントの通知に基づいて実オブジェクト管理情報を自動更新する。

【選択図】 図 1

認定・付加情報

特許出願の番号	平成11年 特許願 第260355号
受付番号	59900895189
書類名	特許願
担当官	濱谷 よし子 1614
作成日	平成11年10月27日

<認定情報・付加情報>

【特許出願人】

【識別番号】	390009531
【住所又は居所】	アメリカ合衆国10504、ニューヨーク州 アーモンク (番地なし)
【氏名又は名称】	インターナショナル・ビジネス・マシーンズ・コーポレーション

【代理人】

【識別番号】	100086243
【住所又は居所】	神奈川県大和市下鶴間1623番地14 日本アイ・ビー・エム株式会社 大和事業所内
【氏名又は名称】	坂口 博

【復代理人】

【識別番号】	申請人
【識別番号】	100104880
【住所又は居所】	東京都港区赤坂7-10-9 第4文成ビル202 セリオ国際特許事務所
【氏名又は名称】	古部 次郎

【選任した代理人】

【識別番号】	100091568
【住所又は居所】	神奈川県大和市下鶴間1623番地14 日本アイ・ビー・エム株式会社 大和事業所内
【氏名又は名称】	市位 嘉宏

【選任した復代理人】

【識別番号】	100100077
【住所又は居所】	東京都港区赤坂7-10-9 第4文成ビル202 セリオ国際特許事務所
【氏名又は名称】	大場 充

出 願 人 履 歴 情 報

識別番号 [390009531]

1. 変更年月日 1990年10月24日

[変更理由] 新規登録

住 所 アメリカ合衆国10504、ニューヨーク州 アーモンク (番地なし)

氏 名 インターナショナル・ビジネス・マシーンズ・コーポレイション